AIX-MARSEILLE UNIVERSITÉ
ED 184 – MATHEMATIQUES ET INFORMATIQUE
LABORATOIRE D'INFORMATIQUE ET DES SYSTEMES

Thèse présentée pour obtenir le grade universitaire de docteur

Discipline : ED 184 MATHEMATIQUES ET INFORMATIQUE
Spécialité : Informatique

Florian BRIDOUX

Simulations intrinsèques et complexités dans les réseaux d'automates

Intrinsic simulations and complexities in automata networks

Soutenue le 15/07/2019 devant le jury composé de :

Jarkko KARI	PR Univ. Turku (Finlande)	Rapporteur
Éric RÉMILA	PR Univ. Saint-Étienne	Rapporteur
Julio ARACENA	PR Univ. Concepción	Examinateur
Emmanuel JEANDEL	PR Univ. Lorraine	Examinateur
Anne SIEGEL	DR CNRS à l'IRISA à Rennes	Examinateur
Adrien RICHARD	CR CNRS au CMM (Chili)	Directeur de thèse
Sylvain SENÉ	PR Univ. Aix-Marseille	Directeur de thèse
Guillaume THEYSSIER	CR CNRS	Directeur de thèse

Résumé

Les objets qui se trouvent au centre de cette thèse sont les systèmes dynamiques discrets, appréhendés à travers les réseaux d'automates finis, qui se définissent comme des vecteurs de fonctions locales de transition associées à chaque automate. Ces réseaux d'automates peuvent aussi bien être vus comme un outil pour modéliser des systèmes d'interactions naturels (réseaux biologiques, réseaux de particules physiques...) et les phénomènes qu'ils induisent que comme modèle de calcul que l'on peut étudier per se. Ces objets sont abordés à travers le prisme de la complexité et selon celui de la simulation (théorique), chacun de ces concepts formant le cœur d'une partie du document.

La première partie est consacrée au rapport entre le graphe d'interaction d'un réseau, qui permet de représenter les influences entre ses automates, et la dynamique sous-jacente. Dans un premier temps, le focus est mis sur le problème des points fixes, ou configurations stables du réseau, qui vise à comprendre les informations que donne un graphe d'interaction sur le nombre de points fixes d'un réseau, et plus précisément sur la complexité algorithmique de ce comptage. Dans un second temps, l'intérêt est porté sur une propriété quelque peu opposée, à savoir l'expansivité. Un réseau est expansif si on peut prédire sa configuration de départ en observant une seule de ses composantes pendant suffisamment longtemps. Autrement dit, un réseau est expansif si sa dynamique est instable et que toute perturbation locale initiale a des répercussions visibles sur chaque automate. Nous caractérisons les graphes d'interactions qui admettent un réseau expansif et étudions la possibilité d'un réseau d'être expansif en fonction de plusieurs paramètres comme la taille de l'alphabet, le « temps d'expansivité » ou le type des fonctions calculées (linéaires, abéliennes...).

La seconde partie est consacrée à la simulation intrinsèque, c'est-à-dire la capacité d'un réseau à contenir la complexité comportementale et la richesse calculatoire d'un autre réseau. Plus précisément, nous nous intéressons à une simulation spécifique fondée sur la capacité d'un réseau à simuler pas à pas toute la dynamique d'un autre réseau. L'un des paramètres sur lesquels l'accent est mis est le mode de mise à jour, qui représente les étapes de temps au cours desquelles les automates du réseau mettent à jour leur état. Il est bien connu que la dynamique d'un réseau dépend fortement des modes de mise à jour. Une question naturelle dans ce contexte est de comprendre quel type de dynamique peut être simulé avec un mode de mise à jour donné. Pour commencer, nous mettons en évidence le fait qu'un réseau mis à jour séquentiellement est moins « puissant » qu'un réseau mis à jour en parallèle. Alors que toute dynamique peut-être simulée par un réseau évoluant en parallèle, nous montrons que pour être simulée séquentiellement, elle peut nécessiter des réseaux plus grands dont nous bornons la taille. Ensuite, nous donnons les caractéristiques de réseaux « complets » dans le sens qu'ils peuvent simuler tous les réseaux d'une taille donnée en faisant varier leurs modes de mise à jour. Nous présentons enfin plusieurs réseaux de taille minimale, ou de « temps de simulation » minimal, et étudions les relations entre ces deux paramètres.

Abstract

The objects at the center of this thesis are discrete dynamical systems, understood through finite automata networks, defined as vectors of local transition functions (so that one function is associated with one automaton). These automata networks can be tools for modeling natural interaction systems (biological networks, particle networks in physics...) and the phenomena they induce. They can also be seen as a model of computation that can be studied per se. These objects are approached through the prism of complexity and (theoretical) simulation, each of these concepts forming the core of a part of the document.

The first part is devoted to the relations between the interaction graph of a network, that represents the influences between its automata, and the underlying dynamics. First, the focus is put on the fixed points problem that aims to understand the information that gives an interaction graph on the number of fixed points of a network, and more precisely on the algorithmic complexity of this counting. Then, our interest is focused on a somewhat opposite property, namely expansiveness. A network is expansive if one can predict its initial configuration by observing only one of its components for a long enough time. In other words, a network is expansive if its dynamics is unstable and if any initial local perturbation has visible repercussions on each automaton. We characterize the interaction graphs that admit an expansive network and study the possibility of a network to be expansive according to several parameters like the size of the alphabet, the "expansiveness time" or the type of the computed functions (linear, abelian...).

The second part is devoted to intrinsic simulation, namely the capacity of a network to contain the behavioral complexity and computational richness of another network. Specifically, we are interested in a specific simulation based on the ability of one network to simulate step by step the dynamics of another network. One of the parameters that is emphasized is the update mode, which represents the time steps in which automata update their state. It is well known that the dynamics of a network strongly depends on update modes. A natural question in this context is to understand what kind of dynamics can be simulated with a given update mode. First, we highlight that a sequentially updated network is less "powerful" than a network updated in parallel. While any dynamics can be simulated by a network evolving in parallel, we show that to be simulated sequentially, it may require larger networks for which we give bounds on the size. Next, we present the characteristics of "complete" networks in the sense that they can simulate all networks of a given size by varying their update modes. Finally, we emphasize several networks of minimum size, or of minimal "simulation time", and study the relations between these two parameters.

Table des matières

Chapitre 1

Introduction

L'objet d'étude de cette thèse est le modèle des *réseaux d'automates*. Un réseau d'automates peut être vu comme un système dynamique composé d'entités, appelées *automates*, évoluant en temps discret et décrivant de manière « minimaliste » les interactions temporelles s'exécutant entre ces entités. Chaque automate possède un état qui appartient à un ensemble fini, et qui peut être mis à jour à l'aide d'une fonction locale qui lui est propre Cette fonction locale dépend de la configuration du système qui est caractérisée par les états courants des automates du réseau. Il est à noter que le terme « automate » utilisé ici désigne simplement une « boite noire qui fait un calcul » et peut être la représentation abstraite d'un grand nombre d'objets : une machine dans un réseau, un neurone, un joueur dans un jeu coopératif, *etc.*

1.1 Présentation générale des réseaux d'automates

1.1.1 Modélisation de phénomènes naturels

Même si le terme « réseau d'automates » n'était pas encore utilisé à l'époque, le modèle d'Ising semble être la première modélisation d'un phénomène naturel à l'aide d'un réseau d'automates. Ce modèle a été créé et utilisé au milieu des années 1920 [82] pour représenter la transition de phase liée au ferromagnétisme, c'est-à-dire le mécanisme grâce auquel certains matériaux dont le fer peuvent être attirés par des aimants ou se transformer en aimant permanent [41]. Dans cette représentation, un automate est une particule et son état, qui peut être positif ($+1$) ou négatif (-1), représente son moment magnétique. Les automates sont organisés selon une grille (généralement en deux dimensions) et l'état d'une particule à un temps $t+1$ dépend du nombre de particules voisines ayant un moment magnétique positif. De nombreuses personnes, comme Ising, Onsager, Ruelle ou Dobrushin ont travaillé ou travaillent encore sur ce modèle et sur la transition de phase lié à celui-ci [50, 51, 115, 136].

Entre les années 1940 et 1970, il y a eu une émergence de travaux visant à modéliser le vivant et des phénomènes biologiques complexes à l'aide de modèles discrets. Notamment, McCulloch et Pitts ont proposé un système neuronal formel dans un article fondateur au début des années 1940 [106]. Ce modèle formel était un réseau d'automates. Cette représentation admet plusieurs hypothèses simplificatrices. Premièrement, dans ce modèle, la structure du réseau de neurones n'évolue pas au cours du temps : le nombre de neurones est figé tout comme les connexions entre ceux-ci. Deuxièmement, un neurone ne peut prendre que deux états : excité (vrai) ou non-excité (faux). À part cet état, il ne conserve aucun souvenir du passé. Troisièmement, les neurones n'ont la possibilité de changer d'état qu'une fois tous les temps Δt (que les auteurs appellent délai synaptique). Ils se mettent alors à jour tous en même temps comme si une horloge leur donnait le tempo. Ceci permet de considérer que l'on a affaire à un temps discret. Enfin, l'état d'excitation d'un neurone à une étape de temps $t+1$ dépend de l'état d'excitation

de ses *voisins* (*i.e.* les neurones auxquels il est relié) à un temps t. On peut voir à quel point, dans cette modélisation, le comportement individuel d'un neurone est simple. Ceci n'empêche pas le modèle d'être très puissant. En effet, il peut calculer n'importe quelle fonction booléenne ce qui le rend par ailleurs Turing-complet si l'on s'autorise un nombre infini de neurones [72].

Ce modèle de réseau de neurones a quelques caractéristiques qu'il partage avec tous les réseaux d'automates. D'abord, la structure globale du réseau est figé, le temps est discrétisé et chaque composante du réseau ne peut prendre qu'un nombre fini d'états (ici deux). En revanche, ils ont par la suite été généralisés de plusieurs manières. Premièrement, le comportement d'un neurone considéré dans ce premier article fait de ce réseau ce que l'on appelle maintenant un réseau à seuil. De nombreux autres types de comportements ont été considérés depuis [44, 112]. Deuxièmement, le synchronisme parfait de la mise à jour des composantes du réseau a souvent été remplacé par d'autres types de mises à jour aussi bien déterministes [60, 73, 74, 152] que non déterministes [71, 78, 157, 158].

À partir de la fin des années 1960, les réseaux d'automates booléens (*i.e.* où les automates n'ont que deux états) ont commencé à être utilisés pour modéliser les réseaux de régulation biologique (notamment génétique), notamment par Kauffman et Thomas [90, 91, 92, 153, 157]. Dans cette représentation, on suppose qu'un gène n'a que deux états : exprimé ou inhibé. L'expression ou la non-expression d'un gène va déclencher ou stopper la production d'un certain nombre de protéines. Enfin, l'évolution de l'expression d'un gène au cours du temps ne dépend que de l'expression ou non d'un certain nombre de ces protéines. Pour simplifier, dans ce modèle, on oublie la notion de protéine. Si par exemple un gène activé déclenche l'émission d'une protéine qui va inhiber un autre gène, on va directement représenter ceci par la fonction locale du deuxième gène qui réagira négativement en fonction de l'état du premier gène. Entre autres résultats, cette modélisation donne une explication de la différenciation cellulaire en interprétant les différents types de cellules soit comme des configurations récurrentes, soit comme des points fixes (configurations stables) du réseau. Des recherches récentes laissent en effet penser que les différents types de cellules observés dans la réalité correspondent à des configurations récurrentes [81, 114] et que certains types de configurations récurrentes normalement inaccessibles donneraient des cellules cancéreuses [120]. On peut citer d'autres travaux en lien avec la biologie. Par exemple, dans [93, 138], les auteurs étudient la dynamique du réseau de régulation génétique de la levure. Dans les articles [47, 94, 95] le modèle est appliqué au problème de la réponse immunitaire.

L'étude d'un système biologique sous le prisme d'un réseau d'automates se heurte à un problème d'information. Certes, les biologistes peuvent obtenir, au moins partiellement, le graphe d'interaction d'un réseau (qu'ils appellent *graphe de régulation*), c'est-à-dire une « carte » des influences entre ses composantes [43]. Ils peuvent également parfois donner des indications sur la nature de cette influence, c'est-à-dire est-ce qu'exciter une composante va plutôt stimuler ou inhiber une autre composante [104]. En revanche, il

est très difficile de deviner la fonction de transition d'un automate (la manière dont il évolue en fonction de son voisinage). De plus, l'ordre de mise à jour est aussi très difficile à cerner alors qu'il peut avoir une importance fondamentale pour la dynamique [130, 131, 132].

En plus de leur utilisation pour modéliser des phénomènes biologiques, les réseaux d'automates ont également été utilisés pour modéliser des interactions sociales comme la propagation d'une opinion [76, 77, 122] ou la ségrégation sociale [139] notamment avec le modèle de Schelling [141].

1.1.2 Modèle de calcul

Comme on vient de le voir, les réseaux d'automates ont surtout émergé comme outil pour modéliser des phénomènes naturels. Cela dit, même si l'on donnera plus tard quelques résultats qui pourraient avoir des applications en modélisation, cette thèse ne porte a priori pas sur la modélisation de phénomènes naturels. À l'inverse, on va voir le réseau d'automates comme un objet mathématique et un modèle de calcul que l'on va étudier *per se*. On peut par exemple considérer les réseaux d'automates comme des modèles de calcul à travers l'analogie avec les programmes. En effet, on peut voir les configurations initiales du système comme les entrées d'un programme, les cycles limites comme les résultats de celui-ci et enfin les trajectoires intermédiaires comme l'exécution de ce programme. Pour défendre cette approche des réseaux d'automates, nous allons maintenant présenter quelques domaines dans lesquels les réseaux d'automates sont utilisés.

Automates cellulaires

Les automates cellulaires peuvent être vus comme des réseaux d'automates infinis avec une condition d'homogénéité particulière. D'abord, le voisinage d'une cellule est régulier. En effet, la définition standard d'un automate cellulaire utilise un groupe pour représenter le voisinage et l'espace via le graphe de Cayley associé. Ensuite, la fonction locale est identique pour tous les automates (appelés ici cellules).

L'un des premiers automates cellulaires est né de discussions entre von Neumann et Ulam à la fin des années 1940 autour de l'autoreproduction et la croissance des cristaux. Le but de von Neumann était alors à fabriquer un système qui avait à la fois la capacité de se répliquer mais aussi d'être universel du point de vue du calcul [110].

L'exemple le plus populaire d'automate cellulaire est surement le jeu de la vie de Conway introduit dans les années 1970 [70]. Dans ce système, chaque automate est réparti sur une grille en deux dimensions et peut prendre 2 états : vivant ou mort. L'état d'une cellule évolue en fonction de son état précédent et de l'état de ses 8 voisins sur la grille (voisinage de Moore). Si une cellule morte a exactement trois cellules voisines

vivantes à un temps t, alors elle devient vivante au temps $t+1$. Sinon elle reste morte. Si une cellule vivante a au plus deux ou au moins quatre voisines vivantes au temps t, alors elle devient morte au temps $t+1$. Sinon elle reste vivante. Malgré ces règles très simples, ce système offre une énorme richesse comportementale et est a été prouvé Turing complet et même intrinsèquement universel [53].

Dans cette thèse, nous allons nous concentrer sur les réseaux d'automates de taille finie (qu'on appellera maintenant simplement « réseaux d'automates »), qui sont très différent des automates cellulaires sur plusieurs points. Par exemple, alors que l'injectivité, la surjectivité et la bijectivité sont équivalentes pour les réseaux d'automates, un automate cellulaire peut être surjectif sans être injectif. Il existe en revanche une propriété qui y ressemble beaucoup qui est énoncée dans le théorème du jardin d'Éden. Ce théorème dit que la pré-injectivité équivaut à la surjectivité si et seulement si le groupe définissant l'espace est moyennable [40]. Bien que les automates cellulaires soient des objets intrinsèquement infinis, ils sont souvent étudiés de manière finie. D'abord, quand on veut simuler numériquement un automate cellulaire pour observer ce qu'il se passe, on doit se restreindre à des représentations finies. L'une des manières de faire cela est alors de considérer des configurations spatialement périodiques ou de fixer des conditions de bords. En outre, certains problèmes sur des systèmes infinis peuvent être étudiés par des approximations finis de plus en plus grandes. Par exemple, un problème ouvert est de savoir si, quand un automate cellulaire est surjectif alors ses configurations temporellement et spatialement périodiques sont denses (voir le problème ouvert 8 de [87]). Or, on peut voir les automates cellulaires restreint à leurs configurations périodiques d'une taille donnée comme des réseaux d'automates (finis).

En plus de leur côté fini qui les différencient déjà des automates cellulaires, les réseaux d'automates que nous allons étudier dans cette thèse n'auront pas nécessairement un espace homogène. Nous verrons qu'à l'instar du théorème du jardin d'Éden dans les automates cellulaires, la structure de l'espace dans les réseaux d'automates joue un rôle très important.

Calcul distribué

Les réseaux d'automates finis peuvent être aussi considérés comme modèle de calcul distribué avec différentes définitions [160, 161].

Les graphes d'automates finis sont des objets cousins des réseaux d'automates [117]. La structure globale d'un graphe d'automates fini est représentée par un graphe fini. Chaque nœud correspond à un automate qui peut prendre un ensemble fini d'états et qui peut envoyer des signaux à ses voisins dans le graphe. Il y a une certaine notion d'homogénéité dans ce modèle car chaque automate calcule la même fonction locale qui dépend de k voisins, au fait prêt que l'on indique à un nœud qui a moins de k voisins que son voisin est « absent ». Ce sont des modèles de calcul distribué où la mémoire limitée des automates et leur absence de connaissance sur la structure globale du réseau

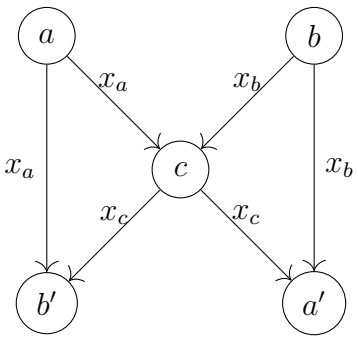
est particulièrement mise en avant. Un problème courant dans ce modèle concerne la reconnaissance par les automates de certaines propriétés sur la structure du graphe.

De manière un peu plus éloignée, le modèle local (souvent noté $\mathcal{LOCAL}$) est un autre modèle cousin des réseaux d'automates où le voisinage de chaque nœud est donné via un graphe. La différence principale de ce modèle avec les réseaux d'automates est que chaque automate a une mémoire du passé mais une capacité de calcul limité. Dans ce modèle, à chaque étape de temps et simultanément, les automates vont envoyer des messages à leurs voisins (de taille arbitraire), recevoir des messages de leurs voisins et faire des calculs à partir des informations reçues. L'exercice classique sur ces objets consiste à faire coopérer les nœuds d'un réseau pour résoudre un problème global comme un problème de coloration par exemple [61, 62].

Donnons maintenant, plusieurs exemples d'utilisation des réseaux d'automates pour résoudre des problèmes concrets.

Network coding

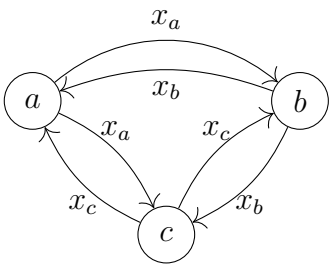
Le *network coding* est un problème de théorie de l'information à la frontière du calcul distribué [80, 89]. L'exemple classique quand on parle de *network coding* est le réseau dit *en papillon* [42] représenté ci-dessous.



On peut voir a,b et c comme des satellites qui envoient des informations (et en reçoit dans le cas de c) et a' et b' comme deux stations au sol qui veulent recevoir un message de la part de a et de b, respectivement. On va considérer que l'information envoyée est binaire (`0` ou `1`) et que les satellites envoient le même message à tous les nœuds qui peuvent les entendre. Le problème consiste à choisir quel message doit être envoyé par chaque satellite pour assurer une transmission correcte. On peut voir que la solution suivante convient. D'abord, les satellites a et b envoient le message x_a et x_b qu'ils veulent envoyer à a' et à b'. Ensuite, le satellite c qui reçoit x_a et x_b envoie un message $x_c = x_a \oplus x_b$. Finalement, les stations a' et b' peuvent déduire respectivement le message x_a et x_b à partir des 2 messages que chacune a reçus. En effet, on a $x_b \oplus x_c = x_b \oplus x_a \oplus x_b = x_a$ et pareillement $x_a \oplus x_c = x_a \oplus x_a \oplus x_b = x_b$.

Cet exemple peut être généralisé. Le nombre de couples satellite/station peut être aussi grand que voulu et le réseau de satellites intermédiaires peut être très compliqué (bien qu'il soit acyclique).

Maintenant, transformons notre problème de la manière suivante : fusionnons chacune des n stations avec le satellite dont elle désire recevoir un message. En reprenant le même exemple on obtient le digraphe *replié* suivant.

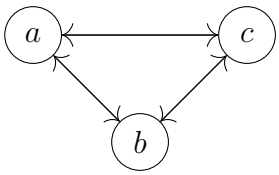


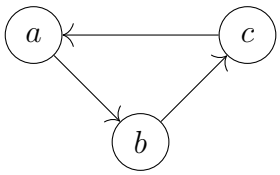
Il a été prouvé que réaliser une telle transmission de message sur le premier digraphe revenait exactement à associer au deuxième digraphe un réseau d'automates sur un alphabet de taille q (*i.e.* chaque automate peut prendre q états différents) avec q^n points fixes (*i.e.* configurations stables) [129]. D'une manière générale, le lien entre les points fixes d'un réseau d'automates et le *network coding* ont été beaucoup étudiés [127, 128].

Exemple du jeu du chapeau

Le jeu du chapeau connait plusieurs variantes mais, dans tous les cas, il s'agit d'un jeu coopératif impliquant n joueurs. Chacun va se voir coiffé d'un chapeau dont il doit deviner la couleur choisie aléatoirement parmi q possibles. Les joueurs ne peuvent en général pas voir leurs propres chapeaux (cela rendrait le jeu trivial). En revanche, ils peuvent voir le chapeau d'autres joueurs selon un certain digraphe.

Par exemple, considérons les deux digraphes suivants.



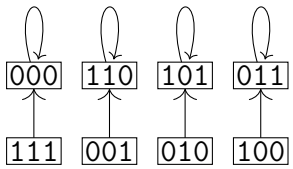


Dans le digraphe de gauche, chaque joueur peut voir la couleur des chapeaux des deux autres joueurs. Dans le digraphe de droite, chaque joueur peut voir un chapeau, par exemple le joueur b peut voir le chapeau du joueur a car il y a un arc entre a et b.

Les joueurs connaissent le digraphe et les règles du jeu. Ils peuvent établir une stratégie commune avant de recevoir leurs chapeaux. En revanche, on considère ici qu'une fois

les chapeaux reçus (tous en même temps), ils ne peuvent pas communiquer et doivent énoncer une couleur au même moment.

Clairement, aucune stratégie ne garantit aux joueurs de trouver la couleur de tous leurs chapeaux en même temps et ce n'est pas ce qui est demandé. La première variante du jeu consiste à demander aux joueurs de maximiser la chance d'avoir raison tous en même temps. Si tous les joueurs répondaient au hasard, ils n'auraient qu'une chance sur q^n d'avoir tous raison en même temps. Considérons le cas où $q = 2$, c'est-à-dire qu'il y a deux couleurs de chapeau (la couleur noire et la couleur blanche par exemple) et où le digraphe utilisé est le digraphe de gauche. Les joueurs peuvent appliquer la stratégie suivante. Chaque joueur compte le nombre de chapeaux noirs qu'il peut voir, et choisit la couleur noire si ce nombre est impair et la couleur blanche sinon. Si le nombre de chapeaux noirs est pair, alors tous les joueurs énonceront la bonne couleur en même temps et sinon ils auront tous tord. Ils ont donc une chance sur deux de gagner. On peut représenter cette solution de la manière suivante.



Ce graphe se lit de la façon suivante. Chaque configuration xyz représente la couleur des chapeaux des joueurs a, b et c respectivement. Le `0` représente un chapeau blanc et un `1` représente un chapeau noir. Chaque configuration xyz est reliée à une configuration $x'y'z'$ qui représente les trois couleurs qui ont été énoncées par les trois joueurs. Si la configuration xyz est égale à la configuration $x'y'z'$ alors les 3 joueurs ont bien deviné leurs couleurs et ont gagné. Sinon, ils ont perdu.

Ce graphe est en fait ce que l'on appelle un graphe de transition et il permet de définir un réseau d'automate. On se rend compte que le problème consiste exactement à trouver un réseau d'automates sur un alphabet de taille q qui maximise le nombre de points fixes (configurations stables) sous la contrainte suivante : la fonction locale qui permet à chaque automate de se mettre à jour ne peut dépendre que des chapeaux que peut observer le joueur correspondant.

Une variante de ce jeu, proposée dans [37], consiste à demander aux joueurs de maximiser non pas la probabilité d'avoir tous raisons mais le nombre de joueurs qui vont donner la bonne couleur dans le pire des cas. En termes de graphe de transition, cela revient à maximiser le nombre de valeurs communes entre une configuration et son image [63].

1.2 Problématiques

Dans cette thèse nous abordons divers problèmes en théorie des réseaux d'automates. Ils sont de natures différentes et font appel à des outils ou techniques spécifiques mais ils sont traversés par trois grandes problématiques que nous introduisons à présent.

Influence du graphe d'interaction

Le graphe d'interaction d'un réseau d'automates permet de représenter de manière très concise les influences entre les différents automates du réseau. Quand il est signé, en plus de spécifier si un automate a une certaine influence sur un autre, il va indiquer si cette influence est positive, négative ou non monotone. Le graphe d'interaction (signé ou non) est schématiquement étudié avec deux approches différentes.

On pourrait qualifier la première approche d'approche de biologiste. En effet, de nombreux biologistes étudient des réseaux de régulation en les modélisant via des réseaux d'automates [45, 56]. Dans ce cadre, le graphe d'interaction (généralement signé) est souvent la principale information que les biologistes sont capables d'acquérir expérimentalement sur le réseau biologique étudié. Le problème consiste alors à étudier les réseaux d'automates qui correspondent exactement à ce graphe d'interaction, car le « vrai » réseau d'automates se trouve parmi eux. En théorie, on pourrait énumérer tous les réseaux compatibles avec ce graphe d'interaction, étudier toutes leurs dynamiques et en déduire toutes les informations possibles sur ce réseau. En pratique, cette approche naïve ne fonctionne que sur des graphes d'interaction de petite taille, car il y a une explosion combinatoire des réseaux d'automates correspondants possibles. Pour cette raison, il y a eu de nombreux travaux théoriques visant à expliciter le rapport entre le graphe d'interaction et la dynamique du réseau. Par exemple, peut-on donner des informations sur la nature des attracteurs en regardant le graphe d'interaction seulement ? On pourrait ajouter à ces questions une nouvelle question peu traitée : quelle est la complexité algorithmique de calculer les propriétés dynamiques d'un réseau à partir de son graphe d'interaction ?

À l'inverse, on pourrait qualifier la seconde approche d'approche d'ingénieur. Dans cette situation, le réseau d'automates n'est pas un objet réel dont on essaie de deviner les caractéristiques mais un objet que l'on fabrique en respectant certaines contraintes. Les deux exemples que l'on a déjà présentés sont les jeux du chapeau et le *network coding*. Dans ces deux cas, les automates ont accès à une information limitée représentée par un graphe. Dans le premier exemple, un joueur ne peut pas tenir compte de la couleur d'un chapeau qu'il ne peut pas voir pour prendre sa décision. Par conséquent, la fonction locale de l'automate qui représente le joueur ne peut dépendre que des voisins de ce joueur dans ce digraphe. En revanche, rien n'interdit à un joueur d'ignorer une information à laquelle il a accès mais qui ne l'intéresse pas. En termes de graphe d'interaction, cela signifie que le graphe d'interaction est un sous-graphe du graphe donné. Cette approche

s'observe également dans les automates cellulaires. En effet, dans un automate cellulaire, le « voisinage » d'une cellule désigne les cellules qui en sont « proches » et que la cellule peut observer. À l'inverse dans les réseaux d'automates, ce qu'on appelle le voisin d'un automate est un autre automate avec lequel il a des interactions. Autrement dit, un réseau d'automates est souvent représenté par un graphe qui représente son essence alors que le graphe donné avec un automate cellulaire représente plus la nature de l'espace.

Influence du mode de mise à jour

La façon dont les automates sont mis à jour au cours du temps a souvent un rôle critique dans la dynamique d'un réseau. Nous pouvons le mettre en évidence avec l'exemple minimaliste suivant (que nous reprendrons à plusieurs reprises dans ce manuscrit). Imaginons un réseau simplement constitué de deux automates a et b, chacun avec une fonction locale qui recopie l'état de l'autre automate. Si les deux automates commençaient dans deux états différents et étaient mis à jour simultanément de manière régulière (ce que l'on appelle un mode de mise à jour parallèle), alors la dynamique du système deviendrait périodique. En effet, après la première série de mises à jour de a et b, les deux automates s'échangeraient leurs valeurs. Après la deuxième série de mises à jour, chacun reprendrait son état original avant de se l'échanger de nouveau à la troisième série de mises à jour, *etc.* Maintenant, imaginons que, sans changer la fréquence d'actualisation, la mise à jour de l'automate a soit légèrement en avance sur celle de b (ce que l'on appelle un mode de mise à jour séquentiel). Alors l'automate a se mettrait dans l'état de b, effaçant à tous jamais l'information de son état de départ et le système entrerait dans une configuration stable (que l'on appelle point fixe) dont il ne pourrait pas sortir (quelle que soit la mise à jour).

En biologie, le mode de mise à jour est souvent perçu comme une incertitude de plus sur la dynamique du système. De nombreux travaux ont étudié l'influence du mode de mise à jour sur la dynamique d'un système [74, 112, 113] et les systèmes « robustes », dont la dynamique n'est pas ou peu affectée par le mode de mise à jour [11]. Une question, que l'on pourrait se poser dans ce contexte est la suivante. Est-ce que l'on peut déduire le mode de mise à jour d'un système à partir de sa dynamique ? Dans d'autres domaines de l'informatique, la question de savoir s'il existe des calculs fondamentalement parallèles est une question ouverte (voir par exemple le problème ouvert « P=NC ? »). Ici, on peut se poser la question suivante. Est-ce qu'il existe des dynamiques intrinsèquement parallèles ? Autrement dit, est-ce qu'il est possible d'observer la dynamique d'un système et de se dire « Ceci n'est pas un système mis à jour de manière séquentielle. » ?

À l'inverse, quand on considère le réseau d'automates comme un objet que l'on peut concevoir et configurer comme on le souhaite pour obtenir une dynamique voulue, le mode de mise à jour peut se transformer en un outil intéressant. En effet, imaginons que l'on dispose d'un circuit imprimé dont chaque composant est un automate que l'on peut

mettre à jour à volonté (avec un bouton poussoir par exemple). Alors, on pourrait « programmer » notre circuit imprimé avec cette série de mises à jour, ce qui permettrait au circuit imprimé de pouvoir calculer plusieurs fonctions différentes. D'ici, on peut se poser plusieurs questions. Quel type de fonctions sont calculables par un tel circuit imprimé ? Que peut-on calculer de cette manière, et avec quelle efficacité ?

On peut analyser cette dernière problématique du point de vue de la simulation intrinsèque, c'est-à-dire la simulation d'un réseau d'automates par un autre réseau d'automates. On peut voir une analogie entre la simulation intrinsèque d'un réseau d'automates et la réduction dans le domaine de la complexité algorithmique. En effet, pour convaincre quelqu'un qu'un problème de décision est difficile, on peut prouver que le résoudre permet de résoudre un autre problème lui-même connu pour être difficile. De la même façon, pour prouver qu'un système est « complexe », on peut montrer qu'il en simule un autre qui lui-même est connu pour être complexe. Par exemple, il est bien connu que la famille des réseaux monotones booléens est universelle. En effet, pour chaque réseau booléen mis à jour en parallèle, on peut trouver un réseau booléen monotone qui le simule pas à pas.

Importance de l'alphabet

Enfin, une autre problématique intéressante, mais peu souvent abordée est celle du nombre d'états d'un réseau d'automates. Plus tôt, on a dit que l'on pouvait schématiquement considérer qu'un gène pouvait avoir deux états : inhibé ou exprimé. Pourtant, des expériences montrent fréquemment des gènes pouvant se trouver dans des états intermédiaires (voir par exemple la figure 3 de l'article de [104]). On pourrait répondre à cette objection que les réseaux booléens sont déjà suffisant à une simulation près. En effet, un automate non booléen peut toujours être représenté par plusieurs automates booléens. Cela dit, on peut se poser la question de la pertinence de cette simplification du point de vue des deux autres problématiques abordées.

D'abord, cette représentation n'a pas forcément les mêmes propriétés de résilience aux variations du mode de mise à jour. En effet, un automate est un objet discret qui ne peut pas être mis à jour « à moitié » alors que sa représentation en deux automates si. Dans le cas extrême, si un réseau d'automate est composé d'un seul automate alors sa dynamique est parfaitement parallèle.

En outre, quand on regarde des propriétés fines (comme la proportion des bassins d'attractions par exemple), un réseau d'automates booléen ne permet pas de simuler par projection n'importe quel réseau d'automates de la même façon qu'on ne peut pas simuler le comportement d'un dé à trois faces en lançant une pièce un nombre fini de fois.

Enfin, quand le graphe d'interaction du réseau est fixé comme dans l'exemple du *network coding*, la taille de l'alphabet peut avoir une importance, et savoir pour quelle taille

d'alphabet on peut obtenir une certaine propriété peut être une question pertinente.

1.3 Organisation du manuscrit

Après un chapitre 2 consacré à fixer des notations et à définir formellement les réseaux d'automates et les outils associés, le manuscrit est partagé en deux parties.

La partie I est consacrée tout entière à l'étude des problématiques liées aux graphes d'interaction des réseaux d'automates.

Dans le chapitre 3, on s'intéressera aux problèmes de déduire le nombre de points fixes d'un réseau admis par un certain graphe d'interaction. C'est un problème très classique mais on l'aborde d'un point de vue original : on veut déterminer la complexité algorithmique de ce problème. On s'attachera à plusieurs variations de ce problème, par exemple en bornant le degré des graphes considérés où en faisant varier le nombre de points fixes recherchés. Selon la version considérée, on verra que le problème peut appartenir à plusieurs classes de complexité (certaines plutôt exotiques) allant de P à NEXPTIME.

Dans le chapitre 4, on s'intéressera à une propriété différente du nombre de points fixes d'un réseau : l'expansivité. Un réseau est expansif si, en partant d'une configuration initiale quelconque, on peut deviner cette configuration initiale simplement en observant l'évolution de n'importe quel automate du réseau pendant suffisamment longtemps. Il y a différentes manières d'interpréter cette propriété. L'une d'entre elle est de se dire que l'on a affaire à un système très chaotique, où la moindre différence locale entre deux configurations va avoir des répercussions globales sur la dynamique du système. Dans ce contexte, on caractérisera les graphes d'interaction qui admettent un réseau expansif sur un certain alphabet. On montrera qu'en plus du graphe d'interaction, une donnée très importante du problème est la *taille de l'alphabet* du réseau. On s'intéressera à des propriétés liées à l'expansivité comme le temps d'expansivité, la fréquence d'expansivité, etc.

La partie II est quant-à-elle consacrée au calcul séquentiel et à la simulation intrinsèque. On va dire dans cette partie qu'un réseau d'automates en simule un autre s'il existe un mode de mise à jour séquentiel du premier réseau (l'état suivant est obtenu par la mise à jour séquentielle des automates selon un ordre prédéterminé) qui produit la même dynamique que le second réseau lorsqu'il est mis à jour en parallèle.

Dans le chapitre 5, on étudiera le problème de la simulation de réseaux mis à jour parallèlement par des réseaux mis à jour en séquentiel. On verra que les réseaux mis à jour séquentiellement ont un « désavantage » pour calculer certaines fonctions : ils doivent être plus grands. On va appeler ce nombre le *coût de séquentialisation* et on va essayer de borner ce coût le plus précisément possible. On mettra particulièrement en valeur le lien entre la taille de l'alphabet et ce coût.

Enfin, le chapitre 6 sera consacré a un cas un peu plus général où on s'autorise à mettre à jour autant de fois que désiré le même automate entre deux étapes de temps. On introduira dans un premier temps les réseaux d'automates *complets*. Ce sont des réseaux d'automates qui peuvent simuler tous les réseaux d'une certaine taille sur un certain alphabet. On construira des réseaux complets de taille minimum ou minimisant le temps de simulation. Enfin, on s'intéressera à la simulation sans mémoire : on considère un réseau mis à jour en parallèle et on veut trouver un réseau qui le simule en étant exactement de la même taille et en travaillant sur le même alphabet. On montrera que même en s'aidant de la répétition, il existe une infinité de réseaux impossibles à simuler dans ces conditions.

Pour clore ce document, on mettra en perspective les résultats obtenus dans le contexte des problématiques évoquées ici.

Chapitre 2

Notations et définitions générales

2.1 Intervalles, mots et permutations

Pour tout $i, j \in \mathbb{N}$, avec $i \leq j$, l'intervalle fermé entre i et j est noté $[i, j] := \{i, i + 1, \ldots, j\}$. De la même façon, on définit les intervalles ouverts et semis-ouvert par $[i, j[:= [i, j] \setminus \{j\}$, $]i, j] := [i, j] \setminus \{i\}$ et $]i, j[:= [i, j] \setminus \{i, j\}$. Quand $n \geq 1$, on utilise $[n]$ comme une abréviation de $[1, n]$.

Un mot $x \in \mathbb{A}^n$ sur un alphabet $\mathbb{A}$ et de taille n peut être vu comme une concaténation de n lettres dans $\mathbb{A}$. Sauf mention contraire, les valeurs des lettres d'un mot x sont indexées de 1 à n. On note donc $x := (x_1, x_2, \ldots, x_n)$. On n'hésitera pas à alléger la notation en enlevant les parenthèses et/ou les virgules quand ça ne posera pas de problèmes de compréhension. On note $\mathbb{A}^*$ l'ensemble des mots de taille finie sur l'alphabet $\mathbb{A}$. Pour tous mots $x = x_1 \ldots x_n$ et $y = y_1 \ldots y_m$, leur concaténation s'écrit $xy := x_1 \ldots x_n y_1 \ldots y_m$. Aussi, x^n est la concaténation de n fois le mot x. En particulier, $(\mathtt{0})^n$ est un mot composé de n fois la lettre $\mathtt{0}$. Pour tout ensemble $I = \{i_1, i_2, \ldots, i_p\} \subseteq [n]$ avec $i_1 < i_2 < \cdots < i_p$, on note $x_I = x_{i_1} x_{i_2} \ldots x_{i_p}$ la projection de x sur I. On note aussi $\mathrm{pr}_I : x \mapsto x_I$ la fonction de projection. On dit qu'un mot y est un *sous-mot* de x s'il existe $I \subseteq [n]$ tel que $x_I = y$. La *distance de Hamming* entre deux mots de même taille $x, y \in \mathbb{A}^n$ est le nombre d'indice $i \in [n]$ tel que $x_i \neq y_i$.

Quand la taille n est claire dans le contexte, on appelle *vecteur de base* i et on note $e^i \in \{\mathtt{0}, \mathtt{1}\}^n$ le mot booléen de taille n tel que $e^i_i = \mathtt{1}$ et pour tout $j \in [n] \setminus \{i\}$, $e^i_j = \mathtt{0}$. De la même façon, pour tout ensemble $I \subseteq [n]$, on note $e^I \in \{\mathtt{0}, \mathtt{1}\}^n$ le mot booléen tel que pour tout $j \in [n]$, $e^I_j = \mathtt{1}$ si $j \in I$ et 0 sinon.

Pour tout $x = (x_1, \ldots, x_n)$ et $y = (y_1, \ldots, y_n)$, deux mots de même taille sur un alphabet $\mathbb{A}$, on notera $x + y = (x_1 + y_1, \ldots, x_n + y_n)$ l'addition composante par composante des deux mots. La nature de l'addition dépendra du contexte, mais ce sera généralement la simple addition modulo q avec q la taille de l'alphabet sur lequel x et y auront été définis.

Pour tout $b \in \{\mathtt{0}, \mathtt{1}\}$, on note $\neg b = \mathtt{0}$ si $b = \mathtt{1}$ et $\mathtt{1}$ sinon.

Pour tout ensemble fini I, on écrit $\mathbb{S}_I$ l'ensemble des *permutations de* I, c'est-à-dire l'ensemble des mots de $I^{|I|}$ sans répétition de lettre. Considérons une permutation $w \in \mathbb{S}_I$. Pour tout $i \in I$, on note $w(i)$ l'indice $t \in [|I|]$ tel que $w_t = i$.

2.2 Réseaux d'automates, fonctions locales et configurations

Cette thèse porte entièrement sur les réseaux d'automates finis (que nous désignerons par « réseaux d'automates » ou simplement « réseaux » dans la suite du document pour

simplifier l'écriture), aussi appelés systèmes dynamiques finis. Un réseau d'automate peut être vu comme une simple fonction $f : \mathbb{A}^n \to \mathbb{A}^n$ avec $\mathbb{A}$ un alphabet de taille q qui est l'ensemble des états dans lesquelles chaque automate peut se trouver et avec n la taille du réseau (*i.e.* le nombre d'automates). Dans toute la suite de cette thèse, sauf mention contraire, n et q sont des entiers avec $n \geq 1$, $q \geq 2$ et $\mathbb{A} := [0, q[$ est un alphabet de taille q. On note $\mathrm{F}(n, q)$, l'ensemble des réseaux de taille n sur l'alphabet $\mathbb{A}$. Quand l'alphabet est de taille 2, on parle de *réseaux d'automates booléens* ou simplement de *réseaux booléens*. On écrit alors $\mathrm{F}(n)$ pour abréger la notation $\mathrm{F}(n, 2)$. Une coordonnée de f est un entier $i \in [n]$ et une *configuration* de f est un mot $x \in \mathbb{A}^n$ de taille n sur l'alphabet $\mathbb{A}$.

Notons $\mathrm{F}_*(n, q)$ l'ensemble des fonctions de $\mathbb{A}^n$ dans $\mathbb{A}$. Il est souvent pratique de voir f comme un tuple de n *fonctions locales* $f_1, \ldots, f_n \in \mathrm{F}_*(n, q)$. En d'autres termes, on a pour tout $x \in \mathbb{A}^n$, $f(x) = (f_1(x), \ldots, f_n(x))$. Pour tout ensemble $I \subseteq [n]$, on note également $f_I = \mathrm{pr}_I \circ f$.

On utilise ici l'exposant d'un réseau f de plusieurs façons. D'abord, de manière classique, pour tout $t \in \mathbb{N}$, $f^t := f \circ \cdots \circ f$ est la composition de t fois la fonction f. De manière moins classique, pour toute coordonnée $i \in [n]$, on définit $f^{(i)} : \mathbb{A}^n \to \mathbb{A}^n$ comme

$$f^{(i)}(x) := (x_1, \ldots, x_{i-1}, f_i(x), x_{i+1}, \ldots, x_n),$$

et pour tout mot $w = (w_1, w_2, \ldots, w_t) \in [n]^t$, on définit $f^w : \mathbb{A}^n \to \mathbb{A}^n$ comme

$$f^w = f^{(w_t)} \circ \cdots \circ f^{(w_2)} \circ f^{(w_1)}.$$

Si w est le mot vide, on considère que f^w est la fonction identité. Remarquons que pour tout $w = uv$ qui est la concaténation de deux mots u et v sur l'alphabet $[n]$, on a $f^w = f^v \circ f^u$. Deuxièmement, pour tout $I \subseteq [n]$, la fonction $f^I : \mathbb{A}^n \to \mathbb{A}^n$ est définie comme suit. Pour tout $x \in \mathbb{A}^n$, si on a $y = f^I(x)$ alors pour tout $i \in [n]$,

$$y_i := \begin{cases} f_i(x) & \text{si } i \in I, \\ x_i & \text{sinon.} \end{cases}$$

en d'autres termes, $f^I(x)$ est la configuration obtenue à partir de x en mettant à jour les coordonnées de I en même temps. Si $f^{(i)} : \mathbb{A}^n \to \mathbb{A}^n$ est la fonction identité $x \mapsto x$, alors on dit que la fonction locale f_i (qui est alors égale à la projection pr_i) est *triviale*.

Pour tout $i \in [n]$, on note $\mathrm{F}^i(n, q) := \{f^{(i)} \mid f \in \mathrm{F}(n, q)\}$ l'ensemble des fonctions qui ne modifient aucune coordonnée en dehors de i. On appelle une *instruction* une fonction de l'ensemble $\mathrm{F}^*(n, q) := \bigcup_{i \in [n]} \mathrm{F}^i(n, q)$.

De manière similaire, pour tout $I \subseteq [n]$, on note $\mathrm{F}^I(n, q) = \{f^I \mid f \in \mathrm{F}(n, q)\}$.

2.3 Graphes et digraphes

Un *graphe* est un couple $G = (V, E)$. Les éléments de V sont appelés *sommets* du graphe et on dit que G est de *taille* $|V|$. Les éléments de E sont appelés *arêtes* du graphe et sont des *paires* $\{i, j\}$ avec $i, j \in V$. On appelle le *voisinage* (ou ensemble des *voisins*) de i dans G est l'ensemble $\mathcal{N}_G(i) := \{j \in V \mid \{i, j\} \in E\}$.

Un *digraphe* est une couple $G = (V, A)$ avec $A \subseteq V \times V$. Les éléments de V sont appelés sommets ici aussi et la taille de G correspond également à $|V|$. En revanche les éléments de A sont appelés *arcs* du digraphe. Un arc de G est donc un couple (i, j) avec $i \in V$ et $j \in V$. En particulier, un digraphe G peut contenir un certain arc (i, j) sans contenir son symétrique (j, i). Le *voisinage entrant* de j dans G est l'ensemble $\mathcal{N}_G(j) := \{i \in V \mid (i, j) \in A\}$. Le *voisinage sortant* de i dans G est l'ensemble $\mathcal{N}_G^{\text{out}}(i) := \{j \in V \mid (i, j) \in A\}$. Un arc (i, j) est un arc entrant de j et sortant de i.

Quand le graphe (ou digraphe) G utilisé est clair dans le contexte, on peut ne pas le préciser et noter simplement $\mathcal{N}(i)$ et $\mathcal{N}^{\text{out}}(i)$.

Un chemin entre i et j de longueur t dans un graphe (*resp.* digraphe) G est un mot $c = (c_1, \ldots, c_{t+1}) \in V^{t+1}$ avec les caractéristiques suivantes. D'abord, $c_1 = i$ et $c_{t+1} = j$. Ensuite, pour tout $k \in [t]$, $\{c_k, c_{k+1}\}$ est une arête de G (*resp.* (c_k, c_{k+1}) est un arc de G). Si c ne contient pas plusieurs fois le même sommet, on dit que c'est un *chemin élémentaire*.

Pour tout couple de sommets $(i, j) \in V$, on appelle la distance $d_G(i, j)$ de i vers j la longueur du plus court chemin entre i et j dans G.

On dit que G est *fortement connexe* si pour tout couple de sommet $(i, j) \in V^2$, il existe un chemin entre i et j.

Un cycle $c = c_1 \ldots c_{t+1}$ de longueur t d'un graphe ou digraphe G est un chemin de longueur t partant de c_1 et arrivant à lui-même (*i.e.* $c_{t+1} = c_1$). Si pout tout $i < j$ avec $(i, j) \neq (1, t + 1)$ on a $c_i \neq c_j$ alors on dit que c est un *cycle élémentaire*. Par défaut, on parlera simplement de cycle pour parler de cycle élémentaire. On dit qu'un graphe ou digraphe est *acyclique* s'il n'a aucun cycle.

On appelle un *feedback vertex set* d'un graphe ou digraphe G un ensemble de sommets qu'on peut retirer de G pour le rendre acyclique. On appelle une arête $\{i, i\} \in E$ ou un arc $(i, i) \in A$ une *boucle* sur i. Un graphe ou digraphe est boucle-complet s'il possède une boucle sur chaque sommet.

2.4 Graphe d'interaction

L'architecture d'un réseau d'automates $f \in \mathrm{F}(n,q)$ est souvent représentée par un digraphe $G=(V,A)$ appelé *graphe d'interaction* de f. Le graphe d'interaction est un objet important qui justifie le nom de *réseau* d'automates. Le graphe d'interaction G indique les influences entre les différentes composantes du réseau. L'ensemble des sommets de G correspond à l'ensemble des coordonnées de f : $V=[n]$. Chaque couple $(i,j) \in V \times V$, est un arc de G si et seulement si

$$\exists x \in \mathbb{A}^n \mid f_j(x) \neq f_j(x+e^i),$$

l'addition étant faite modulo q. Quand on a $(i,j) \in A$, on dit que l'arc (i,j) est *effectif* dans G. Quand le réseau f a pour graphe d'interaction un digraphe $G=(V,A)$ de taille n, on dit aussi que G *admet* le réseau f. On note $\mathrm{F}(G,q)$ l'ensemble des réseaux de $\mathrm{F}(n,q)$ admis par G. Par ailleurs, on abrège $\mathrm{F}(G,2)$ en $\mathrm{F}(G)$ et on utilisera les notations $\mathrm{F}_i(G,q)=\{f_i \mid f \in \mathrm{F}(G,q)\}$, $\mathrm{F}_I(G,q)=\{f_I \mid f \in \mathrm{F}(G,q)\}$, *etc.* .

Aussi, pour tout réseau $f \in \mathrm{F}(n,q)$, on note $\tilde{f}_i : A^{|\mathrm{N}(i)|} \to A$ la fonction locale f_i restreinte à ses voisins entrants. Autrement dit, pour tout $x \in A^n$

$$\tilde{f}_i(x_{\mathrm{N}_G(i)}) = f_i(x).$$

On peut représenter l'architecture de $f \in \mathrm{F}(n,q)$ de manière plus précise en faisant appel au *graphe d'interaction signé* (GIS). Un GIS est un triplet $G=(V,A,\sigma)$ ou (V,A) est un graphe d'interaction de f et où $\sigma : A \mapsto \{-1,0,1\}$ est la fonction définie comme suit : pour tout arc $a=(i,j) \in A$,

— si pour tout $x \in \mathbb{A}^n$ avec $x_i < q-1$ on a $f_j(x) \leq f_j(x+e^i)$ alors $\sigma(a)=1$;

— si pour tout $x \in \mathbb{A}^n$ avec $x_i < q-1$ on a $f_j(x) \geq f_j(x+e^i)$ alors $\sigma(a)=-1$;

— sinon, $\sigma(a)=0$.

En fait, le signe $+1$ (*resp.* -1) sur l'arc (i,j) représente le fait que la fonction locale de j est localement monotone croissante (*resp.* décroissante) en i. À l'inverse, le signe 0 indique que la fonction locale de j est localement non-monotone en i. Si $\sigma(a)=-1$ on dit que l'arc est négatif (ou que i à une influence négative sur j), si $\sigma(a)=1$ on qu'il est positif (influence positive) et sinon ont dit qu'il est nul. Le signe d'un chemin ou d'un cycle élémentaire est le produit des signes de ses arcs.

2.5 Mode de mise à jour, graphe de transition et dynamique

Pour étudier la dynamique d'un réseau d'automates, on doit d'abord définir un mode de mise à jour qu'on peut représenter de la manière suivante. Entre deux étapes de temps discrètes t et $t+1$ que l'on peut observer, il va y avoir un certain nombre d'étapes intermédiaires pendant lesquelles les coordonnées peuvent éventuellement être mises à jour. Un mode de mise à jour indique la liste des sous-étapes auxquelles chaque coordonnée est mise à jour. Plus précisément, dans cette thèse, on va se concentrer sur les modes de mise à jour *déterministes périodiques*. Un mode de mise à jour déterministe périodique pour un réseau de taille n peut toujours être représenté comme un vecteur $W = (W_1, \dots, W_t)$ avec $W_1, \dots, W_t \subseteq [n]$ (même si, dans certains cas ce n'est pas forcément une manière très succincte de les décrire). La dynamique d'un réseau avec le mode de mise à jour W peut alors être décrite par la fonction $f^W = f^{W_t} \circ f^{W_{t-1}} \circ \cdots \circ f^{W_1}$.

Une sous-famille des modes de mise à jour déterministes périodiques W est la famille des modes de mise à jour *bloc-séquentiels* qui respectent les deux conditions suivantes. D'abord, les ensembles $W_1, \dots, W_t$ sont non nuls et disjoints et ensuite $W_1 \cup \cdots \cup W_t = [n]$. Une autre manière de formuler ceci est de dire que chaque coordonnée est mise à jour une et une seule fois entre deux étapes de temps.

Parmi les modes blocs-séquentiels, le mode de mise à jour le plus classique est probablement le *mode de mise à jour parallèle* (parfois appelé mode de mise à jour *synchrone*) où $t = 1$. En d'autre termes, on a $W = [n]$ et la dynamique du réseau avec ce mode peut simplement être décrite par la fonction f.

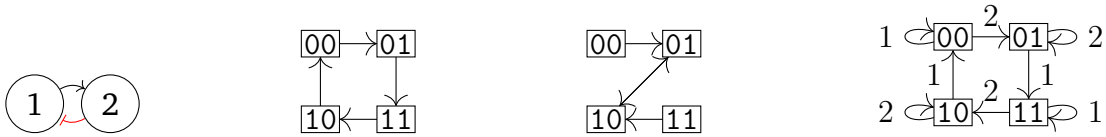
Une autre sous-famille classique de modes de mise à jour blocs-séquentiels est la famille des modes de mise à jour séquentiels où chaque ensemble W_i est de taille 1. Autrement dit, les coordonnées sont mises à jour une à la fois. On représentera chaque mode de mise à jour séquentiel par un mot $w \in S_{[n]}$ tel que $W_1 = \{w_1\}, \dots, W_n = \{w_n\}$. La fonction qui décrit la dynamique de f avec le mode de mise à jour séquentiel w est alors f^w.

Enfin, dans le chapitre 6, on considèrera une généralisation des modes de mise à jour séquentiels qu'on appellera modes de mise à jour séquentiels avec répétition. Dans ce cas-là, entre deux étapes de temps, les coordonnées sont mises à jour une à la fois ($|W_i| = 1$ pour tout i), mais on s'autorise à mettre à jour autant de fois que voulu une même coordonnée. On pourra alors décrire ces modes de mise à jour par des mots $w \in [n]^*$.

Une manière classique de représenter la dynamique d'un réseau $f \in \mathrm{F}(n, q)$ avec un mode de mise à jour W est via son *graphe de transition* $\Gamma_{f,W}$. Le graphe de transition est un digraphe $\Gamma_{f,W} = (\mathbb{A}^n, A)$ où il y a un arc de $x \in \mathbb{A}^n$ vers $y \in \mathbb{A}^n$ si et seulement si $f^W(x) = y$. Notons que chaque configuration a un seul arc sortant. Si W est le mode de mise à jour parallèle, on écrit simplement Γ_f.

Enfin, un autre outil souvent utilisé pour étudier la dynamique d'un réseau f est son *graphe de transition asynchrone* $\tilde{\Gamma}_f$. Il s'agit d'un graphe $\tilde{\Gamma}_f = (\mathbb{A}^n, A)$ tel qu'il existe un arc de $x \in \mathbb{A}^n$ vers $y \in \mathbb{A}^n$ si et seulement s'il existe $i \in [n]$ tel que $f^i(x) = y$. Notons que chaque configuration a au plus n arcs sortants et que tous ses voisins sont à distance de Hamming au plus 1.

En guise d'exemple, considérons le réseau $f \in \mathrm{F}(2)$ défini par $f : x \mapsto (x_2, \neg x_1)$. Sont représentés ci-dessous, dans l'ordre de gauche à droite, son graphe d'interaction signé, et les graphes de transitions Γ_f, $\Gamma_{f,w}$ avec $w = (1, 2)$ et enfin le graphe de transition asynchrone $\tilde{\Gamma}_f$.



Un *cycle limite* est un cycle dans le graphe de transition. Notons qu'il y a entre 1 et q^n cycles dans tout graphe de transition d'un réseau $f \in \mathrm{F}(n, q)$. Une configuration dans un cycle limite est parfois appelée *configuration récurrente*. Un *point fixe* est un cycle limite de taille 1.

Première partie

Graphe d'interaction

Comme nous l'avons vu dans l'introduction, dans l'étude des réseaux d'automates (booléens ou non), le graphe d'interaction (signé ou non) est un objet particulièrement étudié. L'une des raisons est que le graphe d'interaction d'un réseau est souvent la principale information obtenue expérimentalement sur un système réel [100, 156]. Par exemple dans l'article [48], les auteurs donnent le graphe d'interaction représenté figure 2.1.

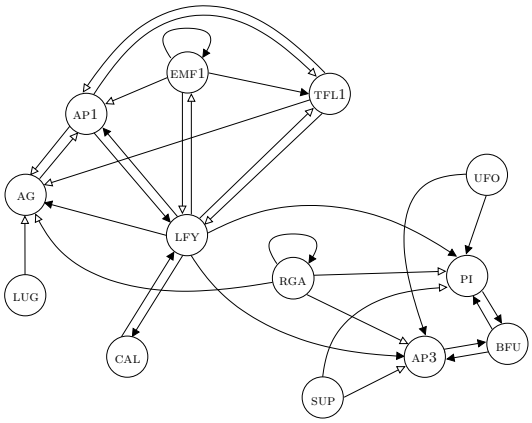


FIGURE 2.1 – Graphe d'interaction du réseau de régulation génétique modélisant l'influence de l'hormone gibbérelline sur la morphogenèse florale de la plante *Arabidopsis thaliana* [48]. Les arcs à flèches pleines (*resp.* creuses) représentent des interactions activatrices (*resp.* inhibitrices).

Parmi les propriétés dynamiques qu'on peut étudier, les points fixes ont un intérêt spécial, car ils correspondent à des motifs stables d'expression génétique à la base de phénotypes cellulaires particuliers [3, 155]. Beaucoup de travaux ont notamment été consacrés à l'étude du nombre de points fixes et à sa maximisation [5, 8, 13, 15, 67, 69, 123, 133].

Les cycles élémentaires (qu'on appellera juste cycles) du graphe d'interaction (signé ou non) d'un réseau d'automates sont connus pour donner beaucoup de renseignements sur sa dynamique. On va présenter quelques-uns d'entre eux. Un résultat bien connu est que si un graphe est acyclique alors sa dynamique est en quelque sorte triviale.

Théorème 1 ([132, 133])**.** *Si le graphe d'interaction G est acyclique, alors pour tout $f \in \mathrm{F}(G, q)$, le graphe de transitions φ_f est composé d'un unique point fixe x, et de chemins vers x de taille au plus n.*

Ceci reste vrai pour tout graphe de transitions $\varphi_{f,W}$ avec W un mode de mise à jour sauf s'il ne met pas à jour certaines coordonnées de f. Pour donner une intuition de preuve, il faut se dire que si le graphe est acyclique, alors il existe $1 \leq n_1 \leq n$ sommets i sans voisins entrant. La fonction locale f_i doit donc être une fonction constante. Par conséquent, quand on met à jour la coordonnée i pour la première fois avec f^W alors la

valeur de l'automate i est fixé pour toujours. Maintenant, si $n_1 = n$ alors la démonstration est terminée. Sinon, il existe $1 \leq n_2 \leq n - n_1$ automates j dont la fonction locale ne dépend que de valeurs déjà fixés. Une fois mises à jour leurs valeurs seront fixées pour toujours. On peut répéter ce raisonnement jusqu'à avoir fixé tous les automates en au plus n étapes.

On rappelle que le signe d'un cycle ou d'un chemin dans un graphe d'interaction signé est le produit des signes de ses arcs. Les deux résultats suivants sont connus dans le cas booléen et facilement généralisable au non booléen.

Théorème 2 ([123]). *Si le graphe d'interaction signé G n'a pas de cycle positifs ou nul, alors pour tout réseau $f \in \mathrm{F}(G, q)$, le graphe de transition φ_f a au plus un point fixe.*

Théorème 3 ([154]). *Si le graphe d'interaction signé G n'a pas de cycle négatif ou nul, alors pour tout réseau $f \in \mathrm{F}(G, q)$, le graphe de transition φ_f a au moins un point fixe.*

On peut montrer qu'on peut borner supérieurement le nombre de points fixes d'un réseau d'automates à partir des cycles de son graphe d'interaction. Plus précisément, on peut le faire grâce aux cycles non négatifs de son graphe d'interaction signé.

Théorème 4 ([5, 9, 124]). *Si tous les cycles positifs ou nuls du graphe d'interaction $G = (V, A)$ passent par $I \subseteq V$ alors pour tout $f \in \mathrm{F}(G, q)$, le graphe de transitions φ_f a au plus $q^{|I|}$ points fixes.*

Chapitre 3

Complexité et points fixes

3.1 Introduction

Ce chapitre porte sur la complexité algorithmique associée à des problèmes de comptage de points fixes. Pour rappel, un point fixe est un cycle limite de taille 1 dans un graphe de transition. Soit $f \in \mathrm{F}(n,q)$ un réseau et W un mode de mise à jour bloc-séquentiel associé. Si la configuration $x \in \mathbb{A}^n$ est un point fixe de $\varphi_{f,W}$ alors cela signifie que $f^W(x) = x$. L'ensemble des points fixes de $\varphi_{f,W}$ ne dépendent en fait que de f et pas de W. En d'autres termes, on a $f^W(x) = x$ si et seulement si $f(x) = x$. Dans ce chapitre, on parlera donc simplement de *points fixes de* f. Notons que l'on ne pourrait pas faire la même chose avec l'ensemble des cycles limites par exemple. En effet, prenons $f : x_1x_2 \mapsto x_2x_1$, un réseau déjà pris comme exemple dans l'introduction. Si on considère le mode de mise à jour parallèle $v = \{1,2\}$, on voit que les configurations $\mathtt{01}$ et $\mathtt{10}$ font partie d'un cycle limite de $\varphi_{f,v}$. En effet,

$$\mathtt{01} \xrightarrow{f} \mathtt{10} \xrightarrow{f} \mathtt{01}.$$

Maintenant, si on considère le mode de mise à jour séquentiel $w = (1,2)$, on voit qu'aucune de ces deux configurations n'appartient à un cycle limite de $\varphi_{f,w}$. En effet,

$$\mathtt{01} \xrightarrow{f^w} \mathtt{11} \xrightarrow{f^w} \mathtt{11} \quad \text{et} \quad \mathtt{10} \xrightarrow{f^w} \mathtt{00} \xrightarrow{f^w} \mathtt{00}.$$

Maintenant, nous allons aborder ce problème avec une approche originale. En effet, la plupart des travaux parlant de complexité des réseaux d'automates se concentrent sur le problème de décision de la forme suivante. Étant donné un réseau d'automates f et une propriété p, quelle est la complexité de décider si la dynamique décrite par f respecte p ? Il est par exemple connu que le problème de savoir si un réseau d'automates a un point fixe est NP-complet en général [97] et dans P pour certaines familles de réseaux d'automates booléens comme les réseaux monotones ou non expansif [58, 75]. Cela-dit, souvent, en pratique, la seule chose que l'on connait d'un RAB f est son GIS. De ce fait, la question suivante parait plus naturel. Étant donné un GIS G et une propriété p, quelle est la complexité de décider si G admet un réseau qui respecte la propriété p ? Ce chapitre portera entièrement sur cette question, qui bien que naturelle n'avait jamais fait l'objet de travaux antérieur. On se restreindra à l'étude des réseaux d'automates booléens mais étendre ces travaux aux réseaux non booléens est une problématique naturelle que l'on abordera en conclusion.

On note $\Phi(f) := \{x \in \{0,1\}^n \mid f(x) = x\}$ l'ensemble des points fixes d'un RAB $f \in \mathrm{F}(n)$ et $\phi(f) := |\Phi(f)|$ la taille de cet ensemble. On note également $\Phi(G) := \{\phi(f) \mid f \in \mathrm{F}(G)\}$. Enfin, on écrit $\Phi^{\max}(G)$ et $\Phi^{\min}(G)$, respectivement l'élément minimum et maximum de $\Phi(G)$. Dans ce chapitre, nous étudions la complexité de décider si $\Phi^{\max} \geq k$ et si $\Phi^{\min}(G) < k$ pour différentes valeurs de k.

Ce chapitre est organisé de la façon suivante. Après la section 3.2 consacrée aux défi-

nitions et notations, nous étudions d'abord le problème $\Phi^{\max}(G) \geq k$ en considérant que l'entier k est fixé.

Nous prouvons dans la section 3.3 qu'étant donné un GIS, décider si $\Phi^{\max}(G) \geq k$ avec $k = 1$ est dans P.

La section 3.4 est consacrée au même problème pour $k \geq 2$. Il est particulièrement intéressant car plusieurs travaux ont déjà été consacrés à la recherche de conditions nécessaires pour l'existence de multiples points fixes et cela, aussi bien dans des modèles discrets que continus [96, 123, 126, 143]. Dans cette section, on prouve que ce problème est NP-complet même quand l'entrée est restreinte aux graphes G dont le degré est borné par une constante $d \geq 2$.

Dans la section 3.5, on considère le cas où k est un argument du problème au même titre que G. On montre que, dans ce cas-là, le problème de décision $\Phi^{\max}(G) \geq k$ est NEXPTIME-complet. On montre également que le problème devient $\mathsf{NP}^{\#\mathsf{P}}$-complet quand on se restreint aux GIS G dont le degré entrant maximum $\Delta(G)$ est borné par une constante $d \geq 2$.

Enfin, la section 3.6 est consacrée au problème de décision $\Phi^{\min} < k$. On montre qu'il est NEXPTIME-complet pour toute constante $k \geq 1$. De plus, on étudie le cas restreint aux GIS de degré entrant borné. On montre que le problème devient alors $\mathsf{NP}^{\mathsf{NP}}$-complet ou $\mathsf{NP}^{\#\mathsf{P}}$-complet selon la nature de k.

Depuis ces résultats, on peut immédiatement déduire des résultats de complexité sur les problèmes de décision duaux $\Phi^{\max}(G) < k$ et $\Phi^{\min}(G) \geq k$. La Table 3.1 donne un récapitulatif des classes d'appartenance des différents problèmes. De plus, ces problèmes sont en fait complets pour leurs classes d'appartenance (à l'exception de ceux dans la classe P).

TABLE 3.1 – Classes de complexité des différents problèmes.

<table>
<tr><th>Problème</th><th>$\Delta(G) \leq d$</th><th>$k = 1$</th><th>$k \geq 2$</th><th>k en paramètre</th></tr>
<tr><td rowspan="2">$\Phi^{\max}(G) \geq k$</td><td>oui</td><td rowspan="4">P</td><td rowspan="2">NP</td><td>$\mathsf{NP}^{\#\mathsf{P}}$</td></tr>
<tr><td>non</td><td>NEXPTIME</td></tr>
<tr><td rowspan="2">$\Phi^{\max}(G) < k$</td><td>oui</td><td rowspan="2">coNP</td><td>$\mathsf{coNP}^{\#\mathsf{P}}$</td></tr>
<tr><td>non</td><td>coNEXPTIME</td></tr>
<tr><td rowspan="2">$\Phi^{\min}(G) < k$</td><td>oui</td><td colspan="2">$\mathsf{NP}^{\mathsf{NP}}$</td><td>$\mathsf{NP}^{\#\mathsf{P}}$</td></tr>
<tr><td>non</td><td colspan="3">NEXPTIME</td></tr>
<tr><td rowspan="2">$\Phi^{\min}(G) \geq k$</td><td>oui</td><td colspan="2">$\mathsf{coNP}^{\mathsf{NP}}$</td><td>$\mathsf{coNP}^{\#\mathsf{P}}$</td></tr>
<tr><td>non</td><td colspan="3">coNEXPTIME</td></tr>
</table>

Les résultats présentés dans ce chapitre, à l'exception de ceux de la section 3.6, ont

fait l'objet d'un papier de conférence qui a été accepté [25]. La section 3.6, quand-à-elle parle du problème du nombre minimum de points fixes et sera incluse dans une version longue qui devrait être soumise prochainement.

3.2 Définitions et notations

3.2.1 Notations relatives aux graphes d'interaction signés

On a déjà défini les graphes d'interaction signés (GIS) dans la section 2.4 mais on va poser ici quelques notations supplémentaires. On rappelle qu'un GIS $G = (V, A, \sigma)$ est un digraphe (V, A) associé à une fonction $\sigma : A \to \{-1, 0, 1\}$ qui donne un signe (négatif, positif ou nul) à chaque arc de G. Ce signe représente *l'influence* qu'a la valeur x_i sur le résultat de $f_j(x)$. On écrit σ_{ij} ou σ_a le signe de l'arc $a = (i, j) \in A$. Pour tout $s \in \{-1, 0, 1\}$ et tout $j \in V$, on définit $\mathcal{N}_G^s(j)$ comme l'ensemble $\{i \in \mathcal{N}_G(j) \mid \sigma_{ij} = s\}$. En outre, pour tout $S \subseteq \{-1, 0, 1\}$, $\mathcal{N}_G^S(j) = \bigcup_{s \in S} \mathcal{N}_G^s(j)$. Les notations précédentes sont illustrées dans la figure 3.1. Un *graphe d'interaction simplement signé* (GISS) est un GIS qui ne contient aucun arc nul (*i.e.* $\mathcal{N}_G^0(j) = \emptyset$ pour tout $j \in V$). Pour $i \in \mathcal{N}_G^{\{-1,1\}}(j)$, on note

$$\tilde{\sigma}_{ij} = \begin{cases} 0 & \text{si } \sigma_{ij} = -1 \\ 1 & \text{si } \sigma_{ij} = +1 \end{cases}.$$

On peut donc écrire $x_j = x_i + \tilde{\sigma}_{ij}$ pour exprimer de manière succincte le fait que

$$x_j = \begin{cases} x_i & \text{si } \sigma_{ij} = +1 \\ \neg x_i & \text{si } \sigma_{ij} = -1 \end{cases}.$$

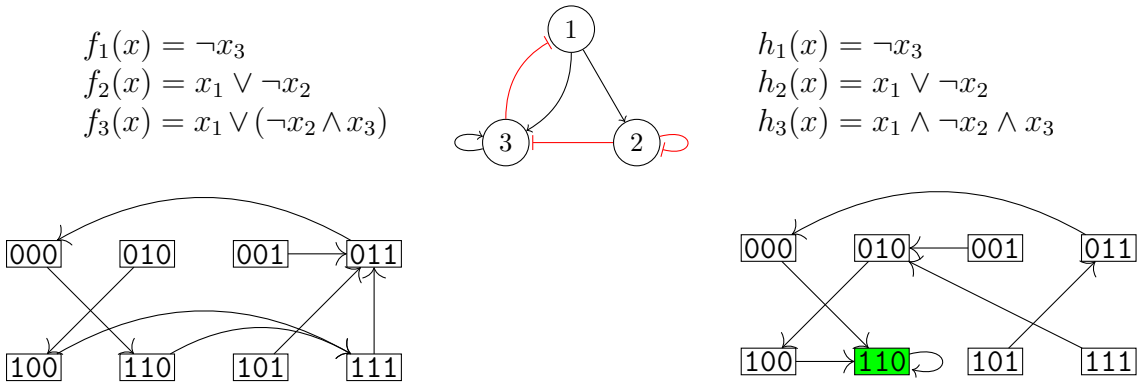


FIGURE 3.1 – Exemple de GISS G avec deux réseaux d'automates booléens $f, h \in \mathrm{F}(G)$. Le RAB f n'a aucun point fixe mais h a le point fixe (110). Il n'y a pas de RAB $g \in \mathrm{F}(G)$ avec plus de points fixes que h. On a donc $\Phi(G) = \{0, 1\}$, $\Phi^{\min}(G) = 0$ et $\Phi^{\max}(G) = 1$. Notons que G a deux cyles positifs et deux cycles négatifs.

Soit $G = (V, A, \sigma)$ un GISS et $f \in \mathrm{F}(G)$. On dit que la fonction locale f_j est la fonction AND si $f_j(x) = \bigwedge_{i \in \mathcal{N}_G(j)} x_i + \tilde{\sigma_{ij}}$. On définit une fonction OR de façon similaire.

Quand le nombre de voisins entrants d'un sommet j du graphe est fortement réduit, le nombre de fonctions contenues dans $\mathrm{F}_j(G)$ l'est aussi. Par exemple, quand j n'a aucun voisin entrant, la fonction $f_j \in \mathrm{F}_j(G)$ correspond à la fonction constante 0 ($x \mapsto 0$) ou par la fonction constante 1 ($x \mapsto 1$). Quand j a un unique arc entrant (i, j) et qu'il est positif (*resp.* négatif), alors $f_j \in \mathrm{F}_j(G)$ est la fonction de copie $x \mapsto x_i$ (*resp.* la fonction négation $x \mapsto \neg x_i$). Quand j a un unique arc entrant nul et strictement moins de deux arcs entrants non nuls alors $\mathrm{F}_j(G) = \emptyset$ [118].

On va maintenant donner dans la proposition 1 quelques résultats de base sur les fonctions locales de $\mathrm{F}_j(G)$.

Proposition 1. *Soit $G = (V, A, \sigma)$ et $j \in V$ tel que $I = \mathcal{N}^{\{-1,1\}}(j)$ est non vide, et soit $I^0 = \mathcal{N}^0(j)$. On a les propriétés suivantes :*

1. *soit $x, y \in \{0, 1\}^{|V|}$ deux configurations telles que $x_{I^0} = y_{I^0}$ et telles que pour tout $i \in I$, $x_i + \tilde{\sigma}_{ij} \leq y_i + \tilde{\sigma}_{ij}$. Alors, $f_j(x) \leq f_j(y)$;*
2. *si $|I^0| \leq 1$ alors, pour tout $x \in \{0, 1\}^{|V|}$, il existe $i \in I$ tel que $x_i + \tilde{\sigma}_{ij} = f_j(x)$.*

Démonstration. Preuve de la la propriété 1. Soit $x, y \in \{0, 1\}^{|V|}$ deux configurations telles que $x_{I^0} = y_{I^0}$ et pour tout $i \in I$, $x_i + \tilde{\sigma}_{ij} \leq y_i + \tilde{\sigma}_{ij}$. Soit $\Delta(x, y)$ l'ensemble des $k \in V$ tels que $x_k \neq y_k$. Remarquons que $\Delta(x, y)$ et I^0 sont disjoints. On procède par induction sur $|\Delta(x, y)|$. Si $|\Delta(x, y)| = 0$ alors on a $x = y$ et donc $f_j(x) = f_j(y)$. Sinon, il existe $k \in \Delta(x, y)$ et on pose $z = x + e^k$. Comme $z_k = y_k$, il est clair que $z_{I^0} = y_{I^0}$ et $z_i + \tilde{\sigma}_{ij} \leq y_i + \tilde{\sigma}_{ij}$ pour tout $i \in I$. De plus, on a $\Delta(z, y) = \Delta(x, y) \setminus \{k\}$, et donc, par hypothèse de récurrence,

$f_j(z) \leq f_j(y)$. Si $k \notin \mathcal{N}(j)$, on a bien $f_j(x) = f_j(z) \leq f_j(y)$. Sinon, $k \in I$ (car $\Delta(x, y)$ et I^0 sont disjoints) et on considère deux cas. Si $\tilde{\sigma}_{kj} = 0$ alors $x_k < y_k = z_k$ par hypothèse, et comme l'arc de k à j est positif, $f_j(x) \leq f_j(z) \leq f_j(y)$. Si $\tilde{\sigma}_{kj} = 1$ alors $x_k > y_k = z_k$ par hypothèse, et comme l'arc de k à j est négatif, on a de nouveau $f_j(x) \leq f_j(z) \leq f_j(y)$. Ceci complète l'étape d'induction.

Preuve de la la propriété 2. Supposons d'abord que $|I^0| = 0$. Supposons par l'absurde qu'il existe une configuration $x \in \{0, 1\}^{|V|}$ tel que $x_i + \tilde{\sigma}_{ij} \neq f_j(x)$ pour tout $i \in I$. Si $f_j(x) = 1$ alors, pour tout $y \in \{0, 1\}^{|V|}$ et $i \in I$ on a $x_i + \tilde{\sigma}_{ij} = 0 \leq y_i + \tilde{\sigma}_{ij}$ et donc $1 = f_j(x) \leq f_j(y)$ d'après la propriété 1. On en déduit que f_j est la fonction constante qui retourne 1, ce qui est absurde puisque $|I| \geq 1$. De même, si $f_j(x) = 0$ alors, pour tout $y \in \{0, 1\}^{|V|}$ et $i \in I$ on a $x_i + \tilde{\sigma}_{ij} = 1 \geq y_i + \tilde{\sigma}_{ij}$ et donc $0 = f_j(x) \geq f_j(y)$ toujours d'après la propriété 1. On en déduit que f_j est la fonction constante qui retourne 0, ce qui est absurde puisque $|I| \geq 1$.

Supposons finalement que $|I^0| = 1$. Soit $\{k\} = I^0$. Supposons par l'absurde qu'il existe une configuration $x \in \{0, 1\}^{|V|}$ telle que $x_i + \tilde{\sigma}_{ij} \neq f_j(x)$ pour tout $i \in I$. Par définition de $\mathcal{N}^0(j)$, pour tout $a, b \in \{0, 1\}$, il existe une configuration $y \in \{0, 1\}^{|V|}$ telle que $y_k = a$ et $f_j(y) \neq f_j(y + e^k) = b$. Soit donc $y \in \{0, 1\}^{|V|}$ tel que

$$y_k = x_k \text{ et } f_j(y) \neq f_j(y + e^k) = f_j(x).$$

Si $f_j(x) = 1$ alors $x_i + \tilde{\sigma}_{ij} = 0 \leq y_i + \tilde{\sigma}_{ij}$ pour tout $i \in I$, et donc $1 = f_j(x) \leq f_j(y)$ d'après la propriété 1. De même, si $f_j(x) = 0$ alors $x_i + \tilde{\sigma}_{ij} = 1 \geq y_i + \tilde{\sigma}_{ij}$ pour tout $i \in I$, et donc $0 = f_j(x) \geq f_j(y)$ d'après la propriété 1. Dans les deux cas, on a $f_j(x) = f_j(y)$, ce qui contredit la définition de y. Par conséquent, pour tout $x \in \{0, 1\}^n$, il existe $i \in I$ tel que $x_i + \tilde{\sigma}_{ij} = f_j(x)$.

□

3.2.2 Problèmes de décision et classe de complexité

Une 3FNC ψ est une formule logique représentée à l'aide de n *variables* $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ et de m *clauses* $\mu = \{\mu_1, \ldots, \mu_m\}$. Chaque clause μ_j est composée de 3 *littéraux* $\mu_{j,1}$, $\mu_{j,2}$, $\mu_{j,3}$. Un littéral $\mu_{j,\gamma}$ est lui-même composé d'une variable $\lambda_i \in \lambda$ et d'une *polarité* $\rho \in \{\bot, \top\}$. Si $\rho = \top$ on dit que λ_i apparait *positivement* dans la clause μ_j et si $\rho = \bot$ on dit qu'il apparait *négativement*. Une *valuation* de λ est une fonction $v : \lambda \to \{\bot, \top\}$. On dit que v *satisfait* la clause μ_j s'il existe un littéral $\mu_{j,\gamma} = (\lambda_i, \rho)$ tel que $v(\lambda_i) = \rho$. On dit que v *satisfait* (ou est une solution de) la 3FNC ψ s'il satisfait toutes les clauses $\mu_j \in \mu$. Par abus de langage et pour simplifier la notation, on écrira souvent $v(\mu_j) = \top$ (*resp.* $v(\psi) = \top$) pour dire que v satisfait μ_j (*resp.* ψ). Sinon, on écrira $v(\mu_j) = \bot$ (*resp.*

$v(\psi) = \bot$). Pour tout $b \in \{0,1\}^n$, on notera $v^b : \lambda \to \{\bot, \top\}$ la valuation telle que

$$v^b(\lambda_i) = \begin{cases} \bot & \text{si } b_i = 0 \\ \top & \text{sinon} \end{cases} \text{ pour tout } i \in [n].$$

Considérons deux valuations $v : \lambda \to \{\bot, \top\}$ et $v' : \lambda' \to \{\bot, \top\}$ avec λ et λ' disjoints. On définit l'*union* de v et v' comme

$$v \cup v' : \lambda_i \mapsto \begin{cases} v(\lambda_i) & \text{si } \lambda_i \in \lambda \\ v'(\lambda_i) & \text{si } \lambda_i \in \lambda' \end{cases}.$$

Un *problème de décision* p est une fonction qui renvoie vrai ou faux. Un élément de l'ensemble de définition de p est appelé une *instance* du problème. On peut présenter sommairement une *classe de complexité* C comme un ensemble de problèmes que l'on peut calculer avec une certaine machine dans certaines conditions. Par exemple, la classe P regroupe les problèmes que l'on peut résoudre sur une machine de Turing déterministe en temps polynomial. La classe NP est identique mais en travaillant sur une machine non déterministe. Pour prouver qu'un problème p appartient à la classe NP on peut donner un algorithme qui calcule p. Cet algorithme doit pouvoir être exécuté sur une machine de Turing non déterministe en temps polynomial. Ceci nous donne une borne supérieure sur la complexité de p. Toutefois, ceci ne nous dit pas que c'est la plus petite classe à laquelle p appartient. Par exemple, p pourrait appartenir à P et donc à NP. Pour prouver que la classe NP est bien représentative de la complexité de p, il faut prouver que p est NP-difficile, c'est-à-dire qu'il est au moins aussi difficile que tous les problèmes de la classe NP. Pour faire cela, le plus souvent, on prouve que l'on peut *réduire* un problème bien connu pour être NP-complet à p. Par exemple, d'après le théorème de Cook, le problème 3-SAT ci-dessous est NP-complet (voir le théorème 8.2 de [116]).

3-SAT
Entrée: Une 3FNC ψ sur les variables $\lambda = \{\lambda_1, \ldots, \lambda_n\}$.
Sortie: Existe-t-il une valuation $v : \lambda \to \{\bot, \top\}$ tel que $v(\psi) = \top$?

Dans ce chapitre, on va rencontrer plusieurs classes de complexité C. On les définira au fur et à mesure en même temps que des problèmes C-difficiles bien connus. Notons que pour de nombreuses classes étudiées ici, le fait que ces classes soient distinctes n'a jamais été prouvé. Par exemple, à l'heure actuelle, le fait que les classes P et NP soient différentes n'est qu'une conjecture.

Dans ce chapitre, on va s'intéresser aux quatre problèmes de décision suivants.

k-MAXIMUM FIXED POINT PROBLEM (k-MAXFPP)
Entrée: Un GIS G.
Sortie: $\Phi^{\max}(G) \geq k$?

MAXIMUM FIXED POINT PROBLEM (MAXFPP)
Entrée: Un GIS G et un entier $k \geq 1$.
Sortie: $\Phi^{\max}(G) \geq k$?

k-MINIMUM FIXED POINT PROBLEM (k-MINFPP)
Entrée: Un GIS G.
Sortie: $\Phi^{\min}(G) < k$?

MINIMUM FIXED POINT PROBLEM (MINFPP)
Entrée: Un GIS G et un entier $k \geq 1$.
Sortie: $\Phi^{\min}(G) < k$?

Une remarque importante concernant la difficulté de ces problèmes est la suivante. Il y a $|\mathrm{F}(n)| = (2^n)^{2^n}$ réseaux d'automates booléens de taille n alors qu'il n'y a que 4^{n^2} GIS de cette taille. En effet, pour chaque couple de sommets $(i, j) \in V \times V$, on peut se demander si ce couple est un arc positif, négatif ou nul du graphe ou si ce n'est pas un arc du graphe. Cela montre que certains GIS correspondent à un nombre exponentiel de réseaux d'automates booléens. Tous les problèmes de décision abordés dans ce chapitre font partie de la classe de complexité NEXPTIME. En d'autres termes, ce sont des problèmes calculables en temps exponentiel sur une machine non déterministe. En effet, par exemple, le problème MAXFPP peut être calculé de la manière suivante. Disons qu'en entrée du problème, on récupère un GIS G et un entier k.

— On choisit de façon non déterministe la fonction $f \in \mathrm{F}(G)$. Cette fonction peut être représentée en binaire en un espace $\log_2(|\mathrm{F}(G)|) \leq n2^n$.

— On calcule $\phi(f)$. Pour cela, on compte le nombre de $x \in \{0, 1\}^n$ tel que $f(x) = x$. Ceci va prendre un temps simplement exponentiel.

— On accepte si $\phi(f) \geq k$.

Une branche non déterministe du programme accepte si et seulement si $\Phi^{\max}(G) \geq k$. Cet algorithme donne le résultat en temps exponentiel sur une machine de Turing non déterministe. Notre problème est donc bien dans NEXPTIME.

En revanche, comme nous allons le voir, on peut raffiner la classe de complexité pour certaines valeurs de k. De plus, on peut se restreindre à certaines familles de GIS, comme la famille des GIS dont le degré maximum est borné par une constante d. En l'occurrence, les GIS de cette sous-famille ont l'avantage de n'admettre qu'un nombre polynomial de réseaux d'automates booléens. En effet, si $f \in \mathrm{F}(G)$, alors chacune de ses fonctions locales $\tilde{f}_i$ est une fonction de $\{0, 1\}^{|\mathcal{N}(i)|}$ dans $\{0, 1\}$. On a $|\mathcal{N}(i)| \leq d$. Il y a donc moins de $c = 2^{2^d}$ choix possibles pour chaque fonction locale. Donc, $|\mathrm{F}(G)| \leq cn$ ce qui est seulement linéaire car c est une constante. Dans certains cas, se restreindre à cette sous-famille diminuera la classe de complexité du problème.

La remarque suivante montre que se restreindre aux GIS G tels que $\Delta(G) \leq 1$ réduit énormément la difficulté du problème. Par la suite on va étudier des degrés entrants maximums inférieurs à une constante $d \geq 2$.

Remarque 1. *Les problèmes de décision* k-MaxFPP, MaxFPP, k-MinFPP *et* MinFPP *restreints aux* GIS G *tels que* $\Delta(G) \leq 1$ *sont tous dans* P.

Démonstration. Soit $G = (V, A, \sigma)$ un GIS avec $\Delta(G) \leq 1$. Le digraphe G est entièrement constitué d'arbres dont les racines sont des cycles ou des sommets sans aucun voisin entrant. Si G contient au moins un arc nul alors $\mathrm{F}(G) = \emptyset$. Sinon, on a $\mathrm{F}(G) = \{f\}$ avec

$$\phi(f) = \begin{cases} 0 & \text{s'il y a un cycle négatif dans } G; \\ 2^\ell & \text{avec } \ell \text{ le nombre de cycles positifs dans } G \text{ sinon.} \end{cases}$$

Calculer $\Phi(G)$ dans ce cas-là se fait donc en temps polynomial. □

3.3 k-Maximum Fixed Point Problem avec $k = 1$

Dans cette section, on s'intéresse au problème de savoir si un GIS G admet une fonction avec au moins un point fixe. C'est le seul problème dont nous allons parler dans ce chapitre dont on ne connait pas la classe de complexité exacte. Plus précisément, le théorème 5 ci-dessous montre que ce problème appartient à P mais nous n'arrivons pas à montrer qu'il est P-difficile.

Théorème 5. 1-MaxFPP *est dans* P.

Le lemme 1 ci-dessous montre que l'on peut très facilement partir d'un GIS G quelconque (*i.e.* potentiellement avec des arcs nuls) et le transformer en un GISS G' tel que $\Phi^{\max}(G) \geq 1$ si et seulement si $\Phi^{\max}(G') \geq 1$. On peut le construire de manière très efficace. De ce fait, pour prouver que 1-MaxFPP est un problème dans P, on pourra considérer que le GIS en entrée du problème est déjà un GISS.

Lemme 1. *Pour tout* GIS G*, il existe un* GISS G' *de même taille tel que* $\Phi^{\max}(G) \geq 1$ *si et seulement si* $\Phi^{\max}(G') \geq 1$.

Démonstration. Construisons $G' = (V, A', \sigma')$ depuis $G = (V, A, \sigma)$. On partitionne l'ensemble V, en quatre ensembles V_{-1}, V_0, V_1, V_2 définis comme suit :

- V_{-1} est l'ensemble des sommets $j \in V$ tel que $|\mathcal{N}^0_G(j)| = 1$ et $|\mathcal{N}_G(j) \setminus \mathcal{N}^0_G(j)| \leq 1$;
- V_0 est l'ensemble des sommets $j \in V$ tel que $|\mathcal{N}^0_G(j)| = 0$;
- V_1 est l'ensemble des sommets $j \in V$ tel que $|\mathcal{N}^0_G(j)| = 1$ et $|\mathcal{N}_G(j) \setminus \mathcal{N}^0_G(j)| \geq 2$;

— V_2 est l'ensemble des sommets $j \in V$ tel que $|\mathcal{N}^0_G(j)| \geq 2$.

On définit G' comme suit. Pour tout $j \in V$:

— si $j \in V_{-1}$ alors l'unique arc entrant du sommet j dans le digraphe G' est une boucle négative ;

— si $j \in V_0$ alors le sommet j a exactement les mêmes arcs entrants dans G et G'. En d'autres termes, $\mathcal{N}^0_{G'}(j) = \emptyset$, $\mathcal{N}^1_{G'}(j) = \mathcal{N}^1_G(j)$ et $\mathcal{N}^{-1}_{G'}(j) = \mathcal{N}^{-1}_G(j)$;

— si $j \in V_1$ alors les arcs entrants du sommet j dans G' correspondent à ceux non nuls dans G. En d'autres termes, $\mathcal{N}_{G'}(j) = \mathcal{N}_G(j) \setminus \{i\}$ avec $\{i\} = \mathcal{N}^0_G(j)$ et pour tout $a \in A'$, $\sigma_a = \sigma_{a'}$;

— si $j \in V_2$ alors le sommet j n'a aucun voisin entrant dans G'.

On a bien défini G'. Montrons maintenant que $\Phi^{\max}(G) \geq 1$ si et seulement si $\Phi^{\max}(G') \geq 1$. Supposons que l'on ait $y \in \{0,1\}^n$ et un réseau $f \in \mathrm{F}(G)$ ou bien un réseau $f' \in \mathrm{F}(G')$ tels que y est un point fixe de f ou f'. Prouvons qu'il existe alors deux réseaux $f \in \mathrm{F}(G)$ et $f' \in \mathrm{F}(G')$ tels que $y \in \Phi(f)$ et $y \in \Phi(f')$. D'abord, on peut se convaincre que l'ensemble V_{-1} est vide. En effet, supposons qu'il existe $j \in V_{-1}$. On voit que $\mathrm{F}_j(G) = \emptyset$ et on a donc $\mathrm{F}(G) = \emptyset$. Il n'existe donc pas de réseau $f \in \mathrm{F}(G)$ tel que $y \in \phi(f)$. De la même façon, $\mathrm{F}_j(G')$ est le singleton $\{f'_j : x \mapsto \neg x_j\}$. Par conséquent, $f'_j(y) = \neg y_j \neq y_j$ et donc y n'est pas un point fixe d'une fonction $f' \in \mathrm{F}(G')$. Donc, si la configuration y est bien définie alors V_{-1} est vide.

Prenons $j \in V_0$. On a alors $\mathrm{F}_j(G) = \mathrm{F}_j(G')$. On peut donc prendre $f_j = f'_j$ et on a $f_j(y) = f'_j(y) = y_j$.

Prenons $j \in V_1$. Soit i tel que $\mathcal{N}^0_G(j) = \{i\}$. D'après la propriété 2 de la proposition 1, si $y \in \Phi(f)$ ou si $y \in \Phi(f')$, il existe $k \in \mathcal{N}_{G'}(j)$ tel que $y_k + \tilde{\sigma}_{kj} = y_j$. Supposons d'abord que $y \in \Phi(f)$. On peut définir f'_j comme la fonction AND si $y_j = 0$ et OR sinon. Dans le premier cas, on a

$$f'_j(y) = (y_k + \tilde{\sigma}_{kj}) \wedge \cdots = y_j \wedge \cdots = 0 \wedge \cdots = 0 = y_j.$$

Dans le deuxième cas, on a

$$f'_j(x) = (y_k + \tilde{\sigma}_{kj}) \vee \cdots = y_j \vee \cdots = 1 \vee \cdots = 1 = y_j.$$

À l'inverse, supposons que $y \in \Phi(f')$. Si $y_j = 0$ on peut définir f_j comme suit :

$$f_j : x \mapsto \Big((x_k + \tilde{\sigma}_{kj}) \vee (x_i + y_i)\Big) \wedge \bigwedge_{\ell \in \mathcal{N}_G(j) \setminus \{k,i\}} \Big((x_\ell + \tilde{\sigma}_{\ell j}) \vee (x_i + \neg y_i)\Big).$$

On remarque que l'ensemble $\mathcal{N}_G(j) \setminus \{k,i\}$ n'est pas vide car $j \in V_1$. On a donc bien $f_j \in \mathrm{F}_j(G)$ et

$$f_j(y) = \Big((y_k + \tilde{\sigma}_{kj}) \vee 0\Big) \wedge \cdots = y_j \wedge \cdots = 0 \wedge = 0 = y_j.$$

De la même façon, si $y_j = \mathtt{1}$, on définit f_j similairement mais en intervertissant les $\wedge$ et les $\vee$ et en remplaçant y_i par $\neg y_i$. On a bien $f_j(y) = \mathtt{1} = y_j$.

Pour finir, prenons $j \in V_2$. Supposons que $y \in \Phi(f)$. On remarque que, comme j n'a aucun voisin entrant dans $\mathcal{N}_{G'}(j)$, l'ensemble $\mathrm{F}_j(G')$ est composé des fonctions constantes $\mathtt{0}$ et $\mathtt{1}$. Il suffit de choisir f'_1 comme la fonction constante y_j et on a bien $f'_j(y) = y_j$. Maintenant, supposons que $y \in \Phi(f')$. Prenons $\mathcal{N}^0_G(j) = \{i_1, \ldots, i_s\}$. Si $y_j = \mathtt{0}$, on peut définir f_j comme suit :

$$f_j : x \mapsto (x_{i_1} + \cdots + x_{i_s} + b) \wedge \bigwedge_{\ell \in \mathcal{N}_G(j) \setminus \mathcal{N}^0_G(j)} (x_\ell + \tilde{\sigma}_{\ell j}),$$

avec b tel que $y_{i_1} + \cdots + y_{i_s} + b = \mathtt{0}$. On a bien, $f_j(y) = (y_{i_1} + \cdots + y_{i_s} + b) \wedge \cdots = \mathtt{0} \wedge \cdots = \mathtt{0} = y_j$. Similairement, si $y_j = \mathtt{1}$, on peut définir f_j de la même façon en substituant les $\wedge$ par des $\vee$ et en prenant b tel que $y_{i_1} + \cdots + y_{i_s} + b = \mathtt{1}$. On a bien $f_j(y) = y_j$.

En conclusion. Depuis un réseau $f \in \mathrm{F}(G)$ tel que $y \in \Phi(f)$, on peut construire le réseau $f' \in \mathrm{F}(G')$ tel que $y \in \Phi(f')$ et inversement. De ce fait, $\Phi^{\max}(G) \geq 1$ si et seulement si $\Phi^{\max}(G') \geq 1$.

□

Remarque 2. *On pourrait aller plus loin et créer depuis G' un* GISS *G'' sans arc positif qui conserverait la propriété du lemme 1. En effet, un arc positif (i, j) de G' peut être remplacé par un sommet supplémentaire k et deux arcs négatifs (i, k) et (k, j) tel que $\mathcal{N}_{G''}(k) = \{i\}$ et $\mathcal{N}^{out}_{G''}(k) = \{j\}$. En fait, remplacer chaque arc positif par deux arcs négatifs ne change pas du tout les nombres de points fixes possibles.*

Soit H une composante fortement connexe d'un GIS G. L'ensemble H est dit *trivial* s'il est composé d'un unique sommet sans aucun arc entrant. De plus, H est dit *initial* s'il n'y a pas d'arc (i, j) tel que $i \notin H$ et $j \in H$.

Le lemme 2 ci-dessous montre que le problème 1-MaxFPP est équivalent à vérifier que chaque composante fortement connexe non triviale et initiale de G contient un cycle positif.

Lemme 2. *Soit $G = (V, A, \sigma)$ un* GISS*. On a l'inégalité $\Phi(G) \neq \{0\}$ si et seulement si chaque composante fortement connexe initiale non triviale de G contient un cycle positif.*

Démonstration. Aracena a déjà prouvé que si un graphe d'interaction signé G a au moins un point fixe alors chaque composante fortement connexe initiale et non triviale H de G contient un cycle positif (voir le corolaire 3 de [5]). Nous en présentons ici une réécriture en utilisant les notations choisies. Soit $f \in \mathrm{F}(G)$, un réseau tel que $\phi(f) \geq 1$. Soit $x \in \Phi(f)$. Notons que chaque sommet $j \in H$ a au moins un voisin entrant et que $\mathcal{N}(j) \subseteq H$. D'après la propriété 2 de la proposition 1, pour tout $j \in H$, il existe un sommet $i \in H$ tel que

$x_i + \tilde{\sigma}_{ij} = x_j$. Partons donc d'un sommet $i_1 \in H$, et trouvons un sommet i_2 tel que $x_{i_2} + \tilde{\sigma}_{i_2 i_1} = x_{i_1}$. De la même façon, on peut trouver un sommet i_3 tel que $x_{i_3} + \tilde{\sigma}_{i_3 i_2} = x_{i_2}$ et répéter l'opération jusqu'à arriver à un sommet i_p qui a déjà été exploré. Cela doit nécessairement se produire en moins de $|H|$ étapes. On a donc un cycle $C = (i_p, \dots, i_t, i_p)$ dans H. Remarquons que tout le long de ce cycle, on a $x_{i_j} = \neg x_{i_{j+1}}$ si et seulement si l'arc (i_j, i_{j+1}) est négatif. Donc, si C a un nombre impair d'arcs négatifs, on a $x_{i_p} = \neg x_{i_p}$. C'est absurde, et on en conclut que C est un cycle positif. Il y a donc bien un cycle positif dans H.

Dans l'autre sens, supposons que chaque composante fortement connexe initiale non triviale de G contienne un cycle positif. On va montrer qu'il existe un réseau $f \in \mathrm{F}(G)$ tel que $\phi(f) \geq 1$. Par ailleurs, on va montrer que l'on peut construire f avec seulement des fonctions locales OR, AND et constantes. Construisons f en même temps qu'un de ses points fixes y. Premièrement, pour chaque i qui est une composante initiale triviale de G, définissons $f_i : x \mapsto \mathtt{0}$ et $y_i = \mathtt{0}$. On a bien $f_i(y) = y_i = \mathtt{0}$. Ensuite, pour chaque composante fortement connexe initiale non triviale H, on considère un cycle positif $C = (c_1, \dots, c_t, c_1)$. On définit f_{c_1} comme la fonction AND, et on pose $y_{c_1} = 0$. Ensuite, pour chaque $i \in [2, t]$, on définit $y_i = y_{c_{i-1}} + \tilde{\sigma}_{c_{i-1}c_i}$. Et pour chaque coordonnée $i \in [2, t]$ telle que $y_{c_i} = \mathtt{0}$ (*resp.* $\mathtt{1}$) on définit f_i comme la fonction AND (*resp.* OR). Si $y_{c_i} = \mathtt{0}$, on a bien,

$$f_{c_i}(y) = (y_{c_{i-1}} + \tilde{\sigma}_{c_{i-1},c_i}) \wedge \dots = \mathtt{0} \wedge \dots = 0 = y_{c_i}.$$

Et si $y_{c_i} = \mathtt{1}$, on a bien,

$$f_{c_i}(y) = (y_{c_{i-1}} + \tilde{\sigma}_{c_{i-1},c_i}) \vee \dots = \mathtt{1} \vee \dots = 1 = y_{c_i}.$$

De plus, C est un cycle positif. Il a donc un nombre pair d'arcs (c_i, c_{i+1}) tels que $\tilde{\sigma}_{c_i,c_{i+1}} = \mathtt{1}$. Par conséquent,

$$\begin{aligned} f_{c_1}(y) &= \left(y_{c_t} + \tilde{\sigma}_{c_t,c_1}\right) \wedge \dots = \left(y_{c_{t-1}} + \tilde{\sigma}_{c_{t-1},c_t} + \tilde{\sigma}_{c_t,c_1}\right) \wedge \dots \\ &= \left(y_{c_1} + \tilde{\sigma}_{c_1,c_2} + \dots + \tilde{\sigma}_{c_t,c_1}\right) \wedge \dots = \left(y_{c_1}\right) \wedge \dots = \left(\mathtt{0}\right) \wedge \dots = \mathtt{0} = y_{c_1}. \end{aligned}$$

Enfin, on peut faire un parcours en profondeur en partant des sommets i pour lesquels on a déjà défini y_i. À chaque étape de ce parcours, on considère un sommet j, et $i \in \mathcal{N}(j)$, son père dans le parcours. On pose $y_j = y_i + \tilde{\sigma}_{ij}$ et fixe f_j comme étant la fonction AND si $y_j = \mathtt{0}$ et OR sinon. Très clairement, $y \in \Phi(f)$, $\phi(f) \geq 1$ et $\Phi(G) \neq \{0\}$. □

Donc, pour décider si $\Phi^{\max}(G) \geq 1$, il suffit de calculer les composantes fortement connexes non triviales initiales de G (ce qui est faisable en temps linéaire [145]) et de vérifier que chacune contient bien un cycle positif. D'après le lemme 1 et la remarque 2, on pourrait se défaire de tous les arcs positifs ou nuls de G. Le problème de trouver un cycle positif dans une composante revient alors au problème de trouver un cycle de longueur paire dans notre nouveau graphe d'interaction. Pendant très longtemps, il était ouvert de

savoir si ce problème était dans P ou dans NP. Mais finalement, un algorithme polynomial a été trouvé indépendamment par Robertson, Seymour et Thomas [135] d'une part et par McCuaig [105] d'autre part.

Théorème 6 ([105, 135])**.** *Il existe un algorithme polynomial pour décider si un digraphe contient un cycle de longueur paire.*

Ceci conclut la preuve du théorème 5 : 1-MaxFPP est bien dans P. À notre connaissance, aucun résultat de difficulté n'a été prouvé pour le problème des cycles pairs. On ne sait donc pas si 1-MaxFPP est complet pour la classe P.

3.4 k-Maximum Fixed Point Problem pour $k \geq 2$

Dans cette section, on s'intéresse au problème k-MaxFPP avec $k \geq 2$. Le théorème 7 ci-dessous montre que le problème est NP-complet.

Théorème 7. *Pour tout $k \geq 2$, k-MaxFPP est NP-complet, même quand on se restreint aux* GIS G *tels que $\Delta(G) \leq 2$.*

On déduit directement le théorème 7 du lemme 3 et du lemme 7.

Prouvons d'abord que le problème appartient à la classe NP.

Lemme 3. *Pour tout $k \geq 2$, k-MaxFPP est dans NP.*

Démonstration. Considérons un GIS $G = (V, A, \sigma)$ avec $V = [n]$. On suppose que $k \leq 2^n$ sinon la réponse est évidemment non. Pour tout $j \in [n]$, définissons $I(j) = \mathcal{N}^{\{-1,1\}}(j)$ et $b^{(j)} \in \{0,1\}^n$ un mot tel que pour tout $i \in I(j)$, $b_i^{(j)} = \tilde{\sigma}_{i,j}$.

Pour vérifier si $\Phi^{\max}(G) \geq k$, on peut exécuter l'algorithme 1.

Dans cet algorithme, la fonction choisir_parmi permet de faire un choix non déterministe. On peut voir que cet algorithme exécuté sur une machine non déterministe s'arrête en un temps d'au plus $\mathcal{O}(n(kn + \Delta(G))^2)$. Ceci est un temps polynomial car k est une constante ici.

Voyons maintenant pourquoi cet algorithme est correct, à savoir, pourquoi une branche non déterministe de l'algorithme renvoie « Vrai » si et seulement si la réponse au problème est « Vrai ».

Supposons d'abord qu'il existe $f \in \mathrm{F}(G)$ avec k points fixes. On peut considérer l'exécution suivante. Premièrement, les k configurations demandées à la ligne 7 vont correspondre à k points fixes distincts de f. Deuxièmement, la configuration x demandée à la ligne 20 correspond à une configuration telle que $x_i = 0$ et $f_j(x) < f_j(x + e^i)$.

Algorithme 1 $\Phi^{\max}(G) \geq k$

Entrée: Un GIS G.

pour $j \in [n]$ **faire**
  $T_j \leftarrow \emptyset$
  $F_j \leftarrow \emptyset$
fin pour
POINTS_FIXES $\leftarrow \emptyset$
pour $p \in [k]$ **faire**
  $x \leftarrow$ choisir_parmi$\Big(\{0,1\}^n \setminus$ POINTS_FIXES$\Big)$.
  POINTS_FIXES $\leftarrow$ POINTS_FIXES $\cup \{x\}$
  pour $j \in [n]$ **faire**
    si $x_j = 0$ **alors**
      $F_j \leftarrow F_j \cup \{x\}$
    sinon
      $T_j \leftarrow T_j \cup \{x\}$
    fin si
  fin pour
fin pour
pour $j \in [n]$ **faire**
  pour $i \in \mathcal{N}(j)$ **faire**
    si $i \in \mathcal{N}^{\{-1,0\}}(j)$ **alors**
      $x \leftarrow$ choisir_parmi$\Big(\{x \in \{0,1\}^n \mid x_i = 0\}\Big)$
      $T_j \leftarrow T_j \cup \{x\}$
      $F_j \leftarrow F_j \cup \{x + e^i\}$
    fin si
    si $i \in \mathcal{N}^{\{0,1\}}(j)$ **alors**
      $x \leftarrow$ choisir_parmi$\Big(\{x \in \{0,1\}^n \mid x_i = 0\}\Big)$
      $F_j \leftarrow F_j \cup \{x\}$
      $T_j \leftarrow T_j \cup \{x + e^i\}$
    fin si
  fin pour
fin pour
pour $j \in [n]$ **faire**
  pour $x \in F$ **faire**
    pour $x' \in T$ **faire**
      si $x_{\mathcal{N}^0(j)} = x'_{\mathcal{N}^0(j)}$ et $(x' + b^{(j)})_{I(j)} \leq (x + b^{(j)})_{I(j)}$ **alors**
        retourne Faux
      fin si
    fin pour
  fin pour
fin pour
retourne Vrai

Cette configuration existe par définition de $\mathcal{N}^{\{-1,0\}}$. De manière similaire, la configuration demandée à la ligne 25 correspond à une configuration x avec $x_i = \mathtt{0}$ qui respecte l'inégalité $f_j(x) > f_j(x + e^i)$. On remarque que pour tout $j \in [n]$ et pour toute configuration $x \in F_j$, on a $f(x) = \mathtt{0}$ et pour tout $x' \in T$, on a $f(x) = \mathtt{1}$. De ce fait et comme $f \in \mathrm{F}(G)$, d'après la première partie de la proposition 1, la condition ligne 34 n'est jamais atteinte. Sur cette exécution, l'algorithme retourne « Vrai ». L'algorithme est donc correct quand la condition $\Phi^{\max}(G) \geq k$ est respectée.

Supposons maintenant que l'algorithme renvoie « Vrai ». Considérons une branche acceptante de l'exécution de notre algorithme. On va définir un réseau $f \in \mathrm{F}(n)$ et prouver qu'il appartient à $\mathrm{F}(G)$. Pour tout $j \in [n]$ et tout $x \in \{\mathtt{0},\mathtt{1}\}^n$, on définit $f_j(x)$ comme suit. S'il existe une configuration $x' \in T_j$ telle que

$$x_{\mathcal{N}^0(j)} = {x'}_{\mathcal{N}^0(j)} \text{ et } (x' + b^{(j)})_{I(j)} \leq (x + b^{(j)})_{I(j)},$$

alors $f_j(x) = \mathtt{1}$. Sinon on fixe $f_j(x) = \mathtt{0}$.

Considérons le GIS $G' = (V, A', \sigma')$ de f et montrons que $G = G'$. Fixons $j \in [n]$ et montrons que pour tout $s \in \{-1, 0, 1\}$, $\mathcal{N}^s_{G'}(j) = \mathcal{N}^s_G(j)$. D'abord, prouvons que $\mathcal{N}_{G'}(j) \subseteq \mathcal{N}_G(j)$. Prenons $i \in \mathcal{N}_{G'}(j)$. Il existe donc une configuration $x \in \{\mathtt{0},\mathtt{1}\}^n$ telle que $f_j(x) = \mathtt{1} \neq \mathtt{0} = f_j(x + e^i)$. Par définition de f_j, il doit exister une configuration $x' \in T_j$ telle que

$$x_{\mathcal{N}^0_G(j)} = {x'}_{\mathcal{N}^0_G(j)} \text{ et } (x' + b^{(j)})_{I(j)} \leq (x + b^{(j)})_{I(j)}.$$

Maintenant, si $i \notin \mathcal{N}_G(j)$, on a également

$$(x + e^i)_{\mathcal{N}^0_G(j)} = {x'}_{\mathcal{N}^0_G(j)} \text{ et } (x' + b^{(j)})_{I(j)} \leq (x + e^i + b^{(j)})_{I(j)}.$$

Donc, $f_j(x + e^i) = \mathtt{1} = f_j(x)$, ce qui contredit notre définition de x. Donc $i \in \mathcal{N}(j)$ et $\mathcal{N}_{G'}(j) \subseteq \mathcal{N}_G(j)$.

On peut voir que grâce aux lignes $20, 21, 22$ et $25, 26, 27$, on a déjà les résultats suivants. Si $i \in \mathcal{N}_G^{\{-1,0\}}(j)$ alors $i \in \mathcal{N}_{G'}^{\{-1,0\}}(j)$ et si $i \in \mathcal{N}_G^{\{0,1\}}(j)$ alors $i \in \mathcal{N}_{G'}^{\{0,1\}}(j)$. Il en résulte que $\mathcal{N}^0_G(j) \subseteq \mathcal{N}^0_{G'}(j)$. Il ne reste plus qu'à prouver que $\mathcal{N}_G^{-1}(j) \cap \mathcal{N}^0_{G'}(j) = \emptyset$ et que $\mathcal{N}^1_G(j) \cap \in \mathcal{N}^0_{G'}(j) = \emptyset$. Prenons $i \in \mathcal{N}_G^{-1}(j)$ et supposons par l'absurde que $i \in \mathcal{N}^0_{G'}(j)$. Cela veut dire qu'il existe une configuration $x \in \{\mathtt{0},\mathtt{1}\}^n$ avec $x_i = \mathtt{0}$ une lettre telle que $\mathtt{0} = f_j(x) < f_j(x + e^i) = \mathtt{1}$. Il existe donc $x' \in t_j$, une configuration telle que

$$(x + e^i)_{\mathcal{N}^0_G(j)} = {x'}_{\mathcal{N}^0_G(j)} \text{ et } (x' + b^{(j)})_{I(j)} \leq (x + e^i + b^{(j)})_{I(j)}.$$

Comme $i \in \mathcal{N}_G^{-1}(j)$, on a $b^{(j)}_\ell = \tilde{\sigma}_{ij} = 1$. Donc,

$$(x' + b^{(j)})_{I(j)} \leq (x + e^i + b^{(j)})_{I(j)} \leq (x + b^{(j)})_{I(j)}.$$

Par conséquent, $f_j(x) = \mathtt{1} = f_j(x + e^i)$ ce qui contredit le fait que $f_j(x) < f_j(x + e^i)$. Donc $i \in \mathcal{N}_G^{-1}(j)$, $\mathcal{N}_G^{-1}(j) \cap \mathcal{N}^0_{G'}(j) = \emptyset$ et $\mathcal{N}_{G'}^{-1}(j) \subseteq \mathcal{N}_G^{-1}(j)$. On peut prouver que $\mathcal{N}^1_{G'}(j) \subseteq \mathcal{N}^1_G(j)$

de manière totalement symétrique.

□

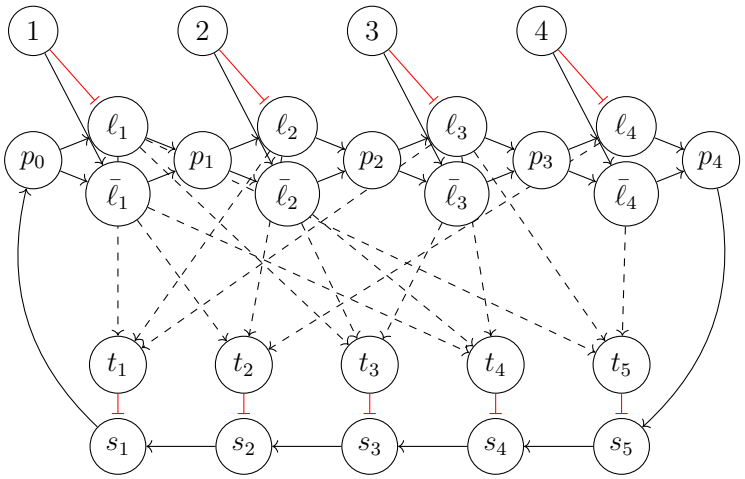


FIGURE 3.2 – GIS G_ψ présenté dans la définition 1. La 3FNC ψ considérée représente la formule suivante :
$(\lambda_1 \vee \lambda_2 \vee \lambda_3) \wedge (\neg\lambda_1 \vee \lambda_2 \vee \lambda_4) \wedge (\lambda_1 \vee \neg\lambda_2 \vee \neg\lambda_3) \wedge (\neg\lambda_1 \vee \neg\lambda_2 \vee \lambda_3) \wedge (\lambda_1 \vee \lambda_3 \vee \neg\lambda_4$
La 3FNC ψ a une solution si et seulement si $\Phi^{\max}(G_\psi) \geq 2$. Sinon $\Phi^{\max}(G_\psi) = 1$. Les arcs noirs avec une pointe en forme de flèche représentent les arcs positifs et ceux en rouge avec une pointe en forme de « T » représentent les arcs négatifs.

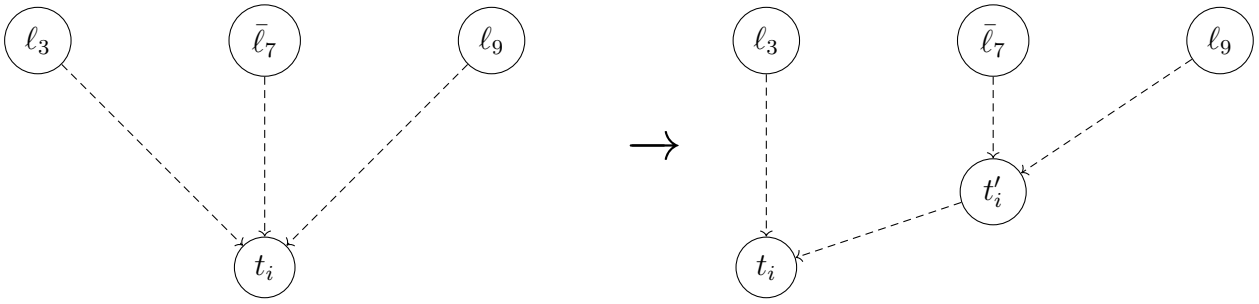


FIGURE 3.3 – Transformation que l'on peut effectuer sur le GIS G_ψ présenté dans la figure 3.2 pour réduire son degré entrant maximum. Cette transformation n'affecte pas les lemmes 5 et 6.

Pour prouver la NP-difficulté du problème 2-MaxFPP, nous allons d'abord définir le GIS G_ψ (illustré dans la figure 3.2). Nous réutiliserons G_ψ plus tard pour prouver la difficulté d'autres problèmes.

Définition 1 (G_ψ)**.** *Soit une* 3FNC ψ *portant sur les* n *variables* $\lambda = \{\lambda_1, \dots, \lambda_n\}$ *et les* m *clauses* $\mu = \{\mu_1, \dots, \mu_m\}$*. On définit le* GIS $G_\psi = (V, A, \sigma)$ *comme suit. D'abord,* $V = R \cup P \cup L \cup \bar{L} \cup S \cup T$ *avec* $R = [n]$, $P = \{p_0, \dots, p_n\}$, $L = \{\ell_1, \dots, \ell_n\}$, $\bar{L} = \{\bar{\ell}_1 \dots, \bar{\ell}_n\}$, $S = \{s_1, \dots, s_m\}$ *et* $T = \{t_1, \dots, t_m\}$*. Notons* $s_0 = p_0$ *et* $s_{m+1} = p_n$*. On définit l'ensemble des arcs* A *comme suit :*

$$\begin{aligned} A := & \bigcup_{i\in[n]} \{(p_{i-1}, \ell_i), (p_{i-1}, \bar{\ell}_i), (\ell_i, p_i), (\bar{\ell}_i, p_i), (i, \ell_i), (i, \bar{\ell}_i)\} \\ & \cup \bigcup_{j\in[m]} \{(t_j, s_j), (s_j, s_{j-1})\} \cup \{(p_n, s_m)\} \\ & \cup \{(\ell_i, t_j) \mid \exists j \in [m] \text{ et } \gamma \in [3] \text{ tels que } \mu_{j,\gamma} = (\lambda_i, \top)\} \\ & \cup \{(\bar{\ell}_i, t_j) \mid \exists j \in [m] \text{ et } \gamma \in [3] \text{ tels que } \mu_{j,\gamma} = (\lambda_i, \bot)\}. \end{aligned}$$

Les arcs dans $\{(s_j, t_j) \mid j \in [m]\} \cup \{(i, \ell_i) \mid i \in [n]\}$ *sont négatifs. Tous les autres sont positifs.*

On note V_ψ l'ensemble des sommets de G_ψ. De plus, on note $\tilde{V}_\psi := V_\psi \setminus [n]$ avec n le nombre de variables dans ψ.

Par la suite, on va beaucoup utiliser la construction G_ψ en lui ajoutant plusieurs sommets et arcs. On définit ci-dessous un ensemble de construction qui diffèrent de G_ψ tout en conservant ses propriétés clés.

Définition 2 (Extension de G_ψ)**.** *On note* $\mathcal{G}_\psi$ *l'ensemble des graphes* $G = (V, A, \sigma)$ *tels que*

- $G_\psi = (V_\psi, A_\psi, \sigma^\psi)$ *est un sous-graphe de* G *(i.e.* $V_\psi \subseteq V, A_\psi \subseteq A$ *et* $\forall a \in A_\psi, \sigma^\psi_a = \sigma_a$*) ;*
- *Les arcs entrants et sortant des sommets de* $\tilde{V}_\psi$ *sont identiques dans* G *et* G_ψ *(i.e. Pour tout* $i \in \tilde{V}_\psi$*,* $\mathcal{N}_G(i) = \mathcal{N}_{G_\psi}(i)$ *et* $\mathcal{N}^{out}_G(i) = \mathcal{N}^{out}_{G_\psi}(i)$*).*

Considérons une 3FNC ψ portant sur les variables $\lambda = \{\lambda_1, \dots, \lambda_n\}$, $G = (V, A, \sigma) \in \mathcal{G}_\psi$ et $f \in \mathrm{F}(G)$.

Pour tout $z \in \{0, 1\}^{|V \setminus \tilde{V}_\psi|}$, soient $\Phi_z(f) := \{x \in \phi(f) \mid x_{V\setminus\tilde{V}_\psi} = z\}$ et $\phi_z(f) = |\Phi_z(f)|$. Pour prouver des résultats sur les extensions de G_ψ on va utiliser le lemme suivant et deux de ses corolaires. Ces résultats ont été prouvés par Julio Aracena [5].

Lemme 4 ([5])**.** *Soit* $G = (V, A, \sigma)$*, un* GIS*,* $f \in \mathrm{F}(G)$ *et* x, y *deux points fixes distincts de* f*. Alors, il existe un cycle positif* C *dans* G *tel que , pour tout arc* (i, j) *dans* C*,* $x_i + \tilde{\sigma}_{ij} = x_j \neq y_j = y_i + \tilde{\sigma}_{ij}$*.*

Corolaire 1. *Supposons que le cycle positif* C *dans le lemme 4 ait seulement des arcs positifs. Alors, soit* $x_i < y_i$ *pour tout sommet* i *dans* C *ou* $x_i > y_i$ *pour tout sommet* i *dans* C*.*

Corolaire 2. *Considérons* $f \in \mathrm{F}(G)$ *et* $\{x, y\} \subseteq \Phi(f)$*. Pour tout ensemble* I *qui soit un* feedback vertex set *de* G*, on a* $x_I \neq y_I$*.*

Le lemme 5 ci-dessous va nous aider à borner inférieurement $\Phi^{\max}(G)$ pour les $G \in \mathcal{G}_\psi$.

Lemme 5. *Soit ψ, une* 3FNC *sur $\lambda = \{\lambda_1, \ldots, \lambda_n\}$, et $G = (V, A, \sigma) \in \mathcal{G}_\psi$. Soit $I = V \setminus \tilde{V}_\psi$. Il existe une fonction $f_{\tilde{V}_\psi} \in \mathrm{F}_{\tilde{V}_\psi}(G)$ telle que*

- *pour tout $f_I \in \mathrm{F}_I(G)$,*
- *pour tout $z \in \{0,1\}^V$, et $b = z_{[n]}$,*

si $f_I(z) = z_I$, alors le réseau f (défini en « concaténant » f_I et $f_{\tilde{V}_\psi}$) respecte

$$\phi_{z_I}(f) \geq \begin{cases} 2 & \text{si } v^b(\psi) = \top, \\ 1 & \text{sinon.} \end{cases}$$

Démonstration. On considère les mêmes ensembles $P, L, \bar{L}, S, T$ que dans la définition 1. Notons $\tilde{V} = \tilde{V}_\psi = P \cup L \cup \bar{L} \cup S \cup T$. On va définir $f_{\tilde{V}} \in \mathrm{F}_{\tilde{V}}(G)$ comme suit.

- Pour tout $i \in L \cup \bar{L} \cup S \cup T$, f_i est la fonction AND.
- Pour tout $i \in P$, f_i est la fonction OR.

Pour tout $a \in \{0, 1\}$, notons $J_a = \{i \in [n] \mid b_i = a\}, L_a = \{\ell_i \in L \mid i \in J_a\}, \bar{L}_a = \{\bar{\ell}_i \in \bar{L} \mid i \in J_a\}$ et $P_a = \{p_i \in P \mid i \in J_a\}$. On définit x et y comme suit.

- $x_I = y_I = z_I$.
- $x_{\tilde{V}} = (0)^{|\tilde{V}|}$.
- $y_{T \cup L_1 \cup \bar{L}_0} = 0 \ldots 0$.
- $y_{S \cup P \cup L_0 \cup \bar{L}_1} = 1 \ldots 1$.

Clairement $x \neq y$. Prouvons maintenant que x est un point fixe de f et que si $v^b(\psi) = \top$ alors y aussi. Par définition de $\mathcal{G}_\psi$, il n'y a aucun arc (i, j) avec $i \in \tilde{V}$ et $j \in I$. Donc, f_I dépend uniquement des coordonnées dans I. Par conséquent, $f_I(x) = f_I(z) = z_I = x_I$ et même chose pour y. D'abord, pour tout $i \in L \cup \bar{L} \cup S \cup T$, i a un prédécesseur j dans $\tilde{V}$ pour lequel $\tilde{\sigma}_{ji} = 0$ et f_i est la fonction AND. Donc,

$$f_i(x) = x_j \wedge \cdots = 0 \wedge \cdots = 0 = x_i.$$

De plus, $f_{p_0}(x) = x_{s_1} = 0 = x_{p_0}$. Et, pour tout $i \in [n]$,

$$f_{p_i}(x) = x_{\ell_i} \vee x_{\bar{\ell}_i} = 0 \vee 0 = 0 = x_{p_i}.$$

Donc, la configuration x est bien un point fixe de f. Maintenant, supposons que $v^b(y) = \top$ avec $b = z_{[n]} = y_{[n]}$.

- Soit $\ell_i \in L_0$. On a $f_{\ell_i}(y) = y_{p_i} \wedge \neg y_i = 1 \wedge \neg 0 = 1 = y_{\ell_i}$.
- Soit $\ell_i \in L_1$. On a $f_{\ell_i}(y) = y_{p_i} \wedge \neg y_i = 1 \wedge \neg 1 = 0 = y_{\ell_i}$

— Soit $\bar{\ell}_i \in \bar{L}_0$. On a $f_{\bar{\ell}_i}(y) = y_{p_i} \wedge y_i = 1 \wedge 0 = 0 = y_{\bar{\ell}_i}$.
— Soit $\bar{\ell}_i \in \bar{L}_1$. On a $f_{\bar{\ell}_i}(y) = y_{p_i} \wedge y_i = 1 \wedge 1 = 1 = y_{\bar{\ell}_i}$.
— $f_{p_0}(y) = y_{s_1} = 1 = y_{p_0}$.
— Soit $p_i \in P_0$. On a $f_{p_i}(y) = y_{\ell_i} \vee y_{\bar{\ell}_i} = 1 \vee 0 = 1 = y_{p_i}$.
— Soit $p_i \in P_1$. On a $f_{p_i}(y) = y_{\ell_i} \vee y_{\bar{\ell}_i} = 0 \vee 1 = 1 = y_{p_i}$.
— Soit $s_j \in S$. On a $f_{s_j}(y) = y_{s_{j+1}} \wedge \neg y_{t_j} = 1 \wedge \neg 0 = 1 = y_{s_j}$.
— Soit $t_j \in T$. Comme $v^b(\psi) = \top$, on sait que la clause μ_j contient un littéral $\mu_{j,\gamma} = (\lambda_i, \rho)$ tel que $v^b(\lambda_i) = \rho$. Il y a deux cas.
  — Si $\rho = \bot$, alors $b_i = 0$, $\bar{\ell}_i \in \bar{L}_0$ et $(\bar{\ell}_i, t_j) \in A$. De plus, $f_{t_j}(y) = y_{\bar{\ell}_i} \wedge \cdots = 0 \wedge \cdots = 0 = y_{t_j}$.
  — Si $\rho = \top$, alors $b_i = 1$, $\ell_i \in L_1$ et $(\ell_i, t_j) \in A$. De plus, $f_{t_j}(y) = y_{\ell_i} \wedge \cdots = 0 \wedge \cdots = 0 = y_{t_j}$.

On a donc $f(y) = y$.

□

Considérons une 3FNC ψ portant sur les variables $\lambda = \{\lambda_1, \ldots, \lambda_n\}$, $G = (V, A, \sigma) \in \mathcal{G}_\psi$ et $f \in \mathrm{F}(G)$. On considère l'ensemble $L = \{\ell_1, \ldots, \ell_n\}$ de la définition 1. On note $\varepsilon^f \in \{0,1\}^n$ la configuration telle pour tout $i \in [n]$, $\varepsilon_i^f = \begin{cases} 0 & \text{si } f_{\ell_i} \text{ est une fonction AND} \\ 1 & \text{sinon} \end{cases}$.

Le lemme 6 ci-dessous va nous aider à borner supérieurement $\Phi^{\max}(G)$ pour tout GIS $G \in \mathcal{G}_\psi$.

Lemme 6. *Soit ψ, une* 3FNC *sur $\lambda = \{\lambda_1, \ldots, \lambda_n\}$, $G = (V, A, \sigma) \in \mathcal{G}_\psi$ et $f \in \mathrm{F}(G)$. Si $z \in \{0,1\}^{|V \setminus \tilde{V}_\psi|}$ et $b = z_{[n]} + \varepsilon^f$, alors,*

$$\phi_z(f) \leq \begin{cases} 2 & \text{si } v^b(\psi) = \top, \\ 1 & \text{sinon.} \end{cases}$$

Démonstration. Premièrement, prouvons que $\phi_z(f) \leq 2$. D'après le corolaire 2, toutes les configurations de $\Phi_z(f)$ diffèrent sur un *feedback vertex set* de G. Cependant, toutes les configurations de $\Phi_z(f)$ ne diffèrent que sur les coordonnées dans $\tilde{V} = \tilde{V}_\psi$. Or, comme $G \in \mathcal{G}_\psi$, tous les cycles de V_ψ passent par p_0. Par conséquent, toutes les configurations dans $\Phi_z(f)$ diffèrent en p_0 et $\phi_z(f) \leq 2$.

Prouvons maintenant que si $\phi_z(f) = 2$ alors v^b est une solution de ψ. Supposons que $\phi_z(f) = 2$. Il existe donc deux configurations $x, y \in \Phi_z(f)$ telles que $x_{p_0} < y_{p_0}$. Comme, $x, y \in \Phi_z(f)$, par définition, $x_{V \setminus \tilde{V}} = y_{V \setminus \tilde{V}} = z$ et x et y diffèrent seulement sur $\tilde{V}$. On remarque qu'aucun cycle positif de G ne passe par un sommet de T. En effet, les cycles qui

passent par T sont négatifs. Donc, d'après le lemme 4 et le corolaire 1, il existe un cycle positif C avec les deux propriétés suivantes :

— le cycle C contient $S \cup P$ et ℓ_i ou $\bar{\ell}_i$ pour tout $i \in [n]$, et

— pour tout $j \in C$, $x_j < y_j$ (car tous les arcs de C sont positifs).

Parmi tous les cycles correspondant à cette définition, on choisit celui qui maximise le nombre de coordonnées i telles que ℓ_i est dans C (*i.e.* si $x_{\ell_i} < y_{\ell_i}$ alors $\ell_i \in C$). On remarque que pour tout $j = \ell_i (resp.\ \bar{\ell}_i)$ (même quand $j \notin C$), on a $x_j \leq y_j$. En effet, f_j ne dépend que des coordonnées p_i et i, on a $x_{p_i} < y_{p_i}$, $x_i = y_i$ et l'arc (p_i, j) est positif.

Affirmation 1. *Pour tout* $i \in [n]$, *on a* $\ell_i \in C \iff b_i = \mathtt{0}$.

Démonstration. D'abord, supposons que $b_i = \varepsilon_i^f + z_i = \mathtt{1}$. Dans ce cas :

— si $z_i = \mathtt{0}$, alors $\varepsilon_i^f = \mathtt{1}$, f_{ℓ_i} est la fonction OR, $x_{\ell_i} = \neg z_i \vee x_{p_i} = \mathtt{1} \vee \mathtt{0} = \mathtt{1}$ et par conséquent, $x_{\ell_i} < y_{\ell_i}$;

— si $z_i = \mathtt{1}$, alors $\varepsilon_i^f = \mathtt{0}$, f_{ℓ_i} est la fonction AND, $y_{\ell_i} = \neg z_i \wedge y_{p_i} = \mathtt{0} \wedge \mathtt{1} = \mathtt{0}$ et par conséquent, $x_{\ell_i} < y_{\ell_i}$.

Par conséquent, si $b_i = \varepsilon_i^f + z_i = \mathtt{1}$ alors $x_{\ell_i} \not< y_{\ell_i}$ et $\ell_i \notin C$.

Ensuite, supposons que $b_i = \varepsilon_i^f + z_i = \mathtt{0}$. Dans ce cas :

— si $z_i = \mathtt{0}$, alors $\varepsilon_i^f = \mathtt{0}$, f_{ℓ_i} est la fonction AND, $x_{\ell_i} = \neg z_i \wedge x_{p_i} = \mathtt{1} \wedge \mathtt{0} = \mathtt{0}$ et $y_{\ell_i} = \neg z_i \wedge y_{p_i} = \mathtt{1} \wedge \mathtt{1} = \mathtt{1}$ et par conséquent $x_{\ell_i} < y_{\ell_i}$;

— si $z_i = \mathtt{1}$, alors $\varepsilon_i^f = \mathtt{1}$, f_{ℓ_i} est la fonction OR, $x_{\ell_i} = \neg z_i \wedge x_{p_i} = \mathtt{0} \vee \mathtt{0} = \mathtt{0}$ et $y_{\ell_i} = \neg z_i \vee y_{p_i} = \mathtt{0} \vee \mathtt{1} = \mathtt{1}$ et par conséquent $x_{\ell_i} < y_{\ell_i}$.

Finalement, si $b_i = \varepsilon^f + z_i = \mathtt{0}$ alors $x_{\ell_i} < y_{\ell_i}$ et $\ell_i \in C$. En effet, on rappelle que si C peut passer par ℓ_i ou $\bar{\ell}_i$, on considère qu'il passe par ℓ_i.

■

On rappelle que l'on note $v^b : \lambda \to \{\bot, \top\}$ la valuation telle que $v^b(\lambda_i) = \begin{cases} \bot & \text{si } b_i = \mathtt{0}, \\ \top & \text{sinon,} \end{cases}$ pour tout $i \in [n]$. Montrons que pour toute clause $\mu_j \in \mu$, on a $v^b(\mu_j) = \top$. Tout d'abord, prouvons que pour tout $t_j \in T$, $x_{t_j} = y_{t_j}$. On sait que $x_{s_j} = \mathtt{0}$ et que $y_{s_j} = \mathtt{1}$. De plus, on sait que pour tout voisin entrant $\ell \in L \cup \bar{L}$ de t_j , on a $x_\ell \leq y_\ell$. Donc, $x_{t_j} \leq y_{t_j}$. Supposons par l'absurde que $x_{t_j} < y_{t_j}$.

Il y a deux cas :

— si f_{s_j} est la fonction OR, on a $x_{s_j} = f_{s_j}(x) = \neg x_{t_j} \vee x_{s_{j+i}} = \mathtt{1} \vee \mathtt{0} = \mathtt{1}$ et donc $x_{s_j} \geq y_{s_j}$ ce qui est une contradiction ;

— si f_{s_j} est la fonction AND, on a $y_{s_j} = f_{s_j}(y) = \neg y_{t_j} \wedge y_{s_{j+i}} = \mathtt{0} \wedge \mathtt{1} = \mathtt{0}$ et donc $x_{s_j} \geq y_{s_j}$ ce qui est une contradiction.

Donc, on a $x_{t_j} = y_{t_j}$. On remarque que tous les voisins entrants de t_j sont dans $L \cup \bar{L}$ et que tous ses arcs entrants sont positifs. On rappelle que pour tout $\ell \in L \cup \bar{L}$, $x_\ell \leq y_\ell$.

Le sommet t_j a donc clairement un voisin entrant ℓ tel que $x_\ell = y_\ell$. Ce voisin entrant ℓ correspond à un littéral $\mu_{j,\gamma} = (\lambda_i, \rho)$. Il y a deux possibilités :

— si $\rho = \top$, alors $\ell = \ell_i \in L$. De plus, comme $x_{\ell_i} = y_{\ell_i}$, alors $\ell_i \notin C$, $v^b(\lambda_i) = \top = \rho$ et d'après l'affirmation 1, $v^b(\mu_j) = \top$;

— si $\rho = \bot$, alors $\ell = \bar{\ell}_i \in \bar{L}$. De plus, comme $x_{\bar{\ell}_i} = y_{\bar{\ell}_i}$, alors $\bar{\ell}_i \notin C$, $\ell_i \in C$ et d'après l'affirmation 1, $v^b(\lambda_i) = \bot = \rho$. Donc, $v^b(\mu_j) = \top$.

Pour toute clause μ_j, on a $v^b(\mu_j) = \top$. Donc, si $\phi_z(f) = 2$ alors v^b est une solution de ψ.

□

Remarque 3. *On peut transformer G_ψ pour que $\Delta(G_\psi)$ soit inférieur à 2 tout en maintenant les lemmes 5 et 6. En effet, seuls les sommets dans T ont un degré entrant supérieur à 3. Comme illustré dans la figure 3.3, on peut réduire ce nombre à deux en ajoutant un sommet intermédiaire. Les démonstrations des lemmes 5 et 6 ne sont quasiment pas affectées par ce changement.*

Lemme 7. *k-MaxFPP avec $k \geq 2$ est NP-difficile, même restreint aux GIS G tels que $\Delta(G) \leq 2$.*

Démonstration. Prouvons d'abord le cas $k = 2$. On réduit 3-SAT à notre problème. Précisément, on va considérer une 3FNC ψ portant sur les variables $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ et les clauses $\mu = \{\mu_1, \ldots, \mu_m\}$. On va montrer que ψ a une solution si et seulement si $\Phi^{\max}(G_\psi) \geq 2$.

D'abord, supposons que $v^b : \lambda \to \{\bot, \top\}$ est une solution de ψ. Montrons qu'il existe un réseau $f \in \mathrm{F}(G_\psi)$ tel que $\phi_b(f) \geq 2$. Très clairement, $G_\psi \in \mathcal{G}_\psi$. Soit $I = V_\psi \setminus \tilde{V}_\psi = [n]$. On définit $f_I \in \mathrm{F}_I(G_\psi)$ comme suit. Pour tout $i \in [n]$, f_i est la fonction constante b_i. Donc, si on prend $z \in \{0,1\}^{|V|}$ avec $z_I = b$, on a $f_I(z) = b = z_I$. D'après le lemme 5, on a donc $\phi_b(f) \geq 2$ et donc $\Phi^{\max}(G_\psi) \geq \phi(f) \geq \phi_b(f) \geq 2$.

Maintenant, prouvons que si $\Phi^{\max}(G_\psi) \geq 2$ alors ψ a une solution v^b. Soit un réseau $f \in \mathrm{F}(G)$ tel que $\phi(f) \geq 2$. On remarque que chaque sommet $i \in [n]$ de G_ψ n'a aucun voisin entrant. Donc, les fonctions f_i avec $i \in [n]$ ne peuvent être que constantes. Soit $b' \in \{0,1\}^n$ la configuration telle que $b'_i = \begin{cases} 0 & \text{si } f_i \text{ est la fonction constante } 0 \\ 1 & \text{si } f_i \text{ est la fonction constante } 1 \end{cases}$. Clairement, pour tout $z \in \Phi(f)$, on a $z_{[n]} = b'$. Par conséquent, $\Phi(f) = \Phi_{b'}(f)$. De plus, d'après le lemme 6, si $\phi_z(f) \geq 2$ alors $v^b(\psi) = \top$ avec $b = b' + \varepsilon^f$. On a montré que la proposition 3-SAT(ψ) est vraie si et seulement si $\Phi^{\max}(G_\psi) \geq 2$. En outre, grâce au lemme 5, on voit que $\Phi^{\max}(G_\psi) = 1$ sinon.

Maintenant, supposons que $k > 2$. Soit $\tilde{k} = \lfloor \log_2(k) \rfloor$ (*i.e.* $2^{\tilde{k}-1} < k \leq 2^{\tilde{k}}$). On construit le graphe G' qui est le graphe G_ψ auquel on ajoute $k-1$ nouveaux sommets isolés avec une boucle positive sur chacun d'eux. Donc, $\Phi^{\max}(G') = 2^{\tilde{k}}$ si ψ a une solution et $2^{\tilde{k}-1}$ sinon. Autrement dit $\Phi^{\max}(G') \geq k$ si et seulement si ψ a une solution.

□

3.5 Maximum Fixed Point Problem

Dans cette section, on va s'intéresser au cas où k n'est plus une constante mais un paramètre du problème. Contrairement aux deux sections précédentes, nous allons voir qu'ici, restreindre $\Delta(G)$ change la complexité du problème.

3.5.1 Avec $\Delta(G)$ borné

Dans cette première sous-section, on va étudier MAXFPP restreint à une famille de GIS dont le degré entrant est borné par une constante. Pour cela, on va introduire plusieurs nouvelles classes de complexité : #P, PP, $\mathsf{P}^{\#\mathsf{P}}$, P^{PP}, $\mathsf{NP}^{\#\mathsf{P}}$ et $\mathsf{NP}^{\mathsf{PP}}$.

La classe #P regroupe les problèmes de comptage des certificats d'un problème de décision qui est dans la classe NP. Par exemple, le problème de décision #P-complet canonique est #SAT qui est défini ci-dessous.

#SAT *Entrée:* Une 3FNC ψ portant sur les variables $\{\lambda_1, \ldots, \lambda_n\}$. *Sortie:* Le nombre de valuations $v : \{\lambda_1, \ldots, \lambda_n\} \to \{\bot, \top\}$ telles que $v(\psi) = \top$.

La classe PP regroupe les problèmes de décision décidés par une machine de Turing probabiliste en temps polynomial avec une probabilité d'erreur inférieure à un demi. Le problème de décision PP-complet canonique est MAJORITY-SAT(MAJSAT) défini ci-dessous.

MAJORITY-SAT (MAJSAT) *Entrée:* Une 3FNC ψ portant sur les variables $\{\lambda_1, \ldots, \lambda_n\}$. *Sortie:* Est-ce que, pour la majorité des valuations $v : \{\lambda_1, \ldots, \lambda_n\} \to \{\bot, \top\}$, on a $v(\psi) = \top$? Autrement dit : #SAT(ψ) $\geq 2^n/2$?

Un *oracle* dans une classe C peut être vu comme une boite noire qui peut répondre à n'importe quel problème de la classe C sans utiliser de ressources. La classe $\mathsf{P}^{\#\mathsf{P}}$ correspond aux problèmes de décision calculables en temps polynomial par une machine de Turing déterministe avec un oracle dans #P. La classe P^{PP} se définit de manière identique mais avec un oracle dans PP cette fois. Les classes $\mathsf{NP}^{\#\mathsf{P}}$ et $\mathsf{NP}^{\mathsf{PP}}$ sont respectivement similaires aux classes $\mathsf{P}^{\#\mathsf{P}}$ et P^{PP} mais avec une machine non déterministe.

Proposition 1. *On a* $\mathsf{NP}^{\#\mathsf{P}} = \mathsf{NP}^{\mathsf{PP}}$.

Démonstration. Il est bien connu que $\mathsf{P}^{\#\mathsf{P}}=\mathsf{P}^{\mathsf{PP}}$ [116]. Supposons que l'on dispose d'un algorithme pour résoudre un problème dans $\mathsf{NP}^{\#\mathsf{P}}$ (*resp.* $\mathsf{NP}^{\mathsf{PP}}$). Chaque appel de l'oracle $\#\mathsf{P}$ (*resp.* PP) répond à un problème dans $\mathsf{P}^{\#\mathsf{P}}$ (*resp.* P^{PP}). On a donc $\mathsf{NP}^{\#\mathsf{P}} \subseteq \mathsf{NP}^{\mathsf{PP}}$ (*resp.* $\mathsf{NP}^{\mathsf{PP}} \subseteq \mathsf{NP}^{\#\mathsf{P}}$) et $\mathsf{NP}^{\#\mathsf{P}} = \mathsf{NP}^{\mathsf{PP}}$. □

Enfin, le problème de décision suivant est connu pour être $\mathsf{NP}^{\mathsf{PP}}$-complet [102] (et donc $\mathsf{NP}^{\#\mathsf{P}}$-complet d'après la proposition 1).

EXISTENTIAL-MAJORITY-3SAT (E-MAJ3SAT)
Entrée: Une 3FNC ψ sur $\{\lambda_1, \ldots, \lambda_n\}$ et un entier $s \in [n]$.
Sortie: Existe-t-il une valuation $v : \{\lambda_1, \ldots, \lambda_s\} \to \{\bot, \top\}$ telle que pour la majorité des valuations $v' : \{\lambda_{s+1}, \ldots, \lambda_n\} \to \{\bot, \top\}$, on a $v \cup v'(\psi) = \top$?

Dans le théorème 8 ci-dessous, on montre que MAXFPP avec $\Delta(G) \leq d$ est $\mathsf{NP}^{\#\mathsf{P}}$-complet.

Théorème 8. *Quand* $\Delta(G) \leq d$, MAXFPP *est* $\mathsf{NP}^{\#\mathsf{P}}$*-complet.*

Le théorème 8 découle directement du lemme 8 qui affirme que le problème appartient à $\mathsf{NP}^{\#\mathsf{P}}$ et du lemme 9 qui affirme que le problème est $\mathsf{NP}^{\#\mathsf{P}}$-difficile.

Lemme 8. *Quand* $\Delta(G) \leq d$, *alors* MAXFPP *est dans* $\mathsf{NP}^{\#\mathsf{P}}$.

Démonstration. L'algorithme suivant est dans $\mathsf{NP}^{\#\mathsf{P}}$ et il résout MAXFPP restreint aux graphes de degré entrant borné par une constante d. On suppose qu'en entrée du problème, on récupère un GIS G et un entier k.

1. On choisit de manière non déterministe les fonctions locales $f_i \in \mathrm{F}_i(G)$ avec $i \in [n]$. On sait qu'il y en a un nombre polynomial en fonction de d.
2. On construit la formule $\psi = (f_1(x) = x_1) \wedge \cdots \wedge (f_n(x) = x_n)$ sur les variables $x_1, \ldots, x_n$. Cette formule est de taille linéaire en fonction de n. En effet, comme noté dans la section 3.2, $f_i(x)$ peut être représentée comme une formule sous forme normale conjonctive de taille constante. De plus, cette dernière peut être transformée en temps constant pour qu'elle exprime $f_i(x) = x_i$.
3. On calcule le nombre de solutions de ψ à l'aide de l'oracle $\#\mathsf{P}$. Ce nombre correspond à $\phi(f)$,
4. On accepte si et seulement si $\phi(f) \geq k$.

Une des branches de cet algorithme non déterministe accepte si et seulement si $\Phi^{\max}(G) \geq k$.

□

Le lemme 9 prétend que quand $\Delta(G) \leq d$, le problème MAXFPP est $\mathsf{NP}^{\#\mathsf{P}}$-difficile.

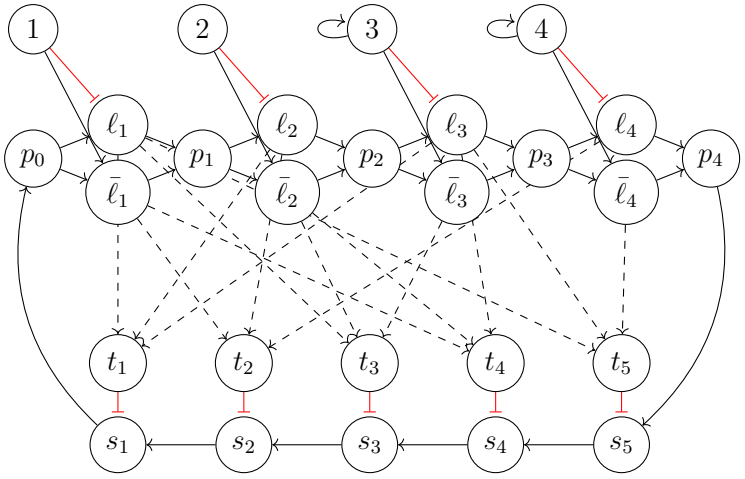


Figure 3.4 – Exemple de réduction de E-Maj3SAT à MaxFPP restreint aux graphes d'interaction G avec un degré entrant borné. Ici est représenté $G_{\psi,2}$ avec ψ la 3FNC présentée dans la figure 3.2. Le problème E-Maj3SAT questionne l'existence de la valuation $v : \{\lambda_1, \lambda_2\} \to \{\bot, \top\}$ telle que la majorité des valuations $v' : \{\lambda_3, \lambda_4\} \to \{\bot, \top\}$ satisfassent ψ. Ce problème a une solution si et seulement si $\Phi^{\max}(G_{\psi,2}) \geq \frac{3}{2}2^2$.

Lemme 9. *Quand $\Delta(G) \leq d$,* MaxFPP *est $\mathsf{NP}^{\#\mathsf{P}}$-difficile.*

Lemme 9. Il suffit de prouver que nous pouvons réduire E-Maj3SAT à MaxFPP. Pour représenter une instance (ψ, s) de E-Maj3SAT, construisons un graphe $G_{\psi,s} = (V, A, \sigma)$ (voir dans la figure 3.4) similaire au graphe G_ψ. La seule différence entre G_ψ et $G_{\psi,s}$ est que l'on ajoute une boucle positive à chacun des $q = n - s$ sommets de $[s+1, n]$. On a donc $G_{\psi,s} \in \mathcal{G}_\psi$. Soit $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ et $\mu = \{\mu_1, \ldots, \mu_m\}$, respectivement les ensembles de variables et de clauses sur lesquelles repose ψ.

Affirmation 2. $\Phi^{\max}(G_{\psi,s}) = 2^{n-s} + \max_{v:\{\lambda_1,\ldots,\lambda_s\}\to\{\bot,\top\}} \alpha(v)$ *avec*

$$\alpha(v) := |\{v' : \{\lambda_{s+1}, \ldots, \lambda_n\} \to \{\bot, \top\} \mid v \cup v'(\psi) = \top\}|.$$

Démonstration. Soit v qui maximise $\alpha(v)$. Prouvons que $\Phi^{\max}(G_{\psi,s}) \geq 2^{n-s} + \alpha(v)$. Soit $a \in \{\mathtt{0}, \mathtt{1}\}^s$, le mot tel que pour tout $i \in [s]$, $v(\lambda_i) = \top$ si et seulement si $a_i = \mathtt{1}$. Soit $I = V_\psi \setminus \tilde{V}_\psi = [n]$. On définit, $f_I \in \mathrm{F}_I(G_\psi)$ comme suit. Pour tout $i \in [s]$, f_i est la fonction constante a_i et pour tout $i \in [s+1, n]$, f_i est la fonction triviale $x \mapsto x_i$. On remarque que pour toute configuration $z \in \{\mathtt{0}, \mathtt{1}\}^{|V|}$ telle que $z_{[s]} = a$ on a $f_I(z) = z_I$. D'après le lemme 5, on peut trouver une fonction $f \in \mathrm{F}(G)$ (qui respecte la définition de f_I déjà donnée) avec la propriété suivante : pour tout $b \in \{\mathtt{0}, \mathtt{1}\}^n$, avec $b_{[s]} = a$, on a $\phi_b(f) \geq 2$ si $v^b(\psi) = \top$ et $\phi_b(f) \geq 1$ sinon. Maintenant, pour chaque valuation $v' : \{\lambda_{s+1}, \ldots, \lambda_n\} \to \{\bot, \top\}$ telle que $v \cup v'(\psi) = \top$ on considère le mot $b = aa'$ tel que v^b corresponde à $v \cup v'$. En d'autres termes, pour tout $i \in [n-s]$, $a'_i = \mathtt{1}$

si et seulement si $v'(\lambda_{i+s}) = \top$. On a bien $b_{[s]} = a$ et $v^b(\psi) = \top$. Par conséquent, $\phi_b(f) \geq 2$. On a donc vu que pour chaque vecteur $b = aa'$, $\phi_b(f) \geq 2$ si le mot a' correspond à une valuation v' telle que $v \cup v'(\psi) = \top$ et que $\phi_b(f) \geq 1$ sinon. On a donc bien $\Phi^{\max}(G_{\psi,s}) \geq 2^{n-s} + \alpha(v)$. En outre, cela montre que $\Phi^{\max}(G_{\psi,s}) \geq 2^{n-s}$.

Maintenant, prouvons qu'il existe une valuation v telle que $\Phi^{\max}(G_\psi) \leq 2^{n-s} + \alpha(v)$. Soit $f \in \mathrm{F}(G)$ un réseau tel que $\phi(f) = \Phi^{\max}(G_{\psi,s})$. On remarque que chaque sommet $i \in [s]$ de $G_{\psi,s}$ n'a aucun voisin entrant. Donc, les fonctions f_i avec $i \in [s]$ ne peuvent être que constantes. On note $a \in \{0,1\}^s$ l'unique image de $f_{[s]}$. De plus, l'unique arc entrant de chaque sommet $i \in [s+1, n]$ de $G_{\psi,s}$ est une boucle positive. Les fonctions f_i avec $i \in [s+1, n]$ sont donc triviales. Remarquons que, pour tout $z \in \Phi(f)$, on a $z_{[s]} = a$. Donc, $\Phi(f) = \bigcup_{a' \in \{0,1\}^{n-s}} \Phi_{aa'}(f)$. De plus, d'après le lemme 6, $\phi_{aa'}(f) \leq 2$ avec égalité si et seulement si $v^b(\psi) = \top$ avec $b = aa' + \varepsilon^f$. Le nombre de mots a' tels que $v^b(\psi) = \top$ avec $b = aa' + \varepsilon^f$ correspond à $\alpha(v)$ avec « $v = v^a$ » (*i.e.* pour tout $i \in [s]$, $v(\lambda_i) = \top$ si et seulement si $a_i = 1$). On a donc $\phi(f) = 2^{n-s} + \alpha(v)$. En conclusion, $\Phi^{\max}(G_\psi) \leq 2^{n-s} + \alpha(v)$.

■

Considérons une valuation $v : \{\lambda_1, \ldots, \lambda_s\} \to \{\bot, \top\}$ qui maximise $\alpha(v)$. La majorité des valuations $v' : \{\lambda_{s+1}, \ldots, \lambda_n\}$ respectent $v \cup v'(\psi) = \top$ si et seulement $\alpha(v) \geq \frac{2^{n-s}}{2}$. Et cela arrive si et seulement si $\Phi^{\max}(G_{\psi,s}) \geq 2^{n-s} + \alpha(v) = \frac{3}{2}2^{n-s}$. On en conclut que E-Maj3SAT(ψ, s) est équivalent à MaxFPP$(G_{\psi,s}, \frac{3}{2}2^{n-s})$.

En utilisant la même astuce que dans la remarque 3, on peut transformer $G_{\psi,s}$ pour que $\Delta(G_{\psi,s}) \leq 2$.

□

3.5.2 Sans restriction sur $\Delta(G)$

Dans la deuxième partie de cette section, nous étudions MaxFPP sans restriction sur le degré entrant maximum des GIS.

Théorème 9. MaxFPP *en général est* NEXPTIME*-complet.*

Démonstration. Comme souligné dans la section 3.2, le problème est dans NEXPTIME. Ci-dessous, on va donc seulement prouver sa NEXPTIME-difficulté.

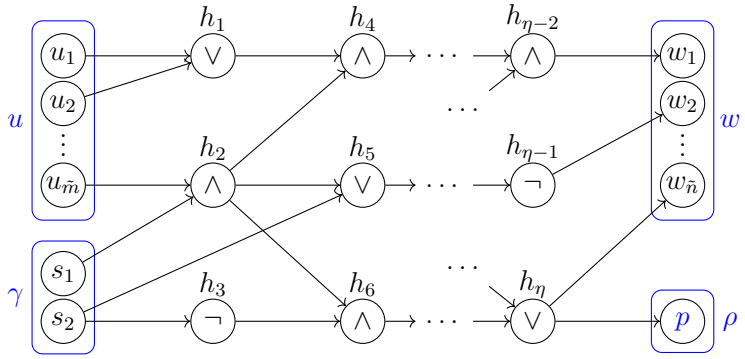


Figure 3.5 – Un circuit C qui encode une 3FNC ψ, instance d'un problème 3-SAT avec $2^{\tilde{m}}$ clauses et $2^{\tilde{n}}$ variables.

Pour prouver que le problème est NEXPTIME-difficile, nous allons faire une réduction depuis Succint-3SAT. Le problème Succint-3SAT est en effet connu pour être NEXPTIME-difficile (voir le théorème 20.2 de [116]).

Soit C, un circuit qui encode une 3FNC Ψ avec $2^{\tilde{n}}$ variables $\Lambda = \{\Lambda_1, \ldots, \Lambda_{2^{\tilde{n}}}\}$ et $2^{\tilde{m}}$ clauses $\mathrm{M} = \{\mathrm{M}_1, \ldots, \mathrm{M}_{2^{\tilde{m}}}\}$. Pour alléger la notation, pour tout $w \in \{0,1\}^{\tilde{n}}$, on pourra écrire Λ_w pour désigner la variable Λ_i avec i le nombre encodé dans w. De la même façon, on parlera de la clause M_u avec $u \in \{0,1\}^{\tilde{m}}$. On parlera aussi du littéral $\mathrm{M}_{u,\gamma}$ avec $\gamma \in \{0,1\}^2$ qui encode une position entre 1 et 3. Le circuit C (représenté dans la figure 3.5) est défini pour encoder la 3FNC Ψ comme suit. Étant donné l'entrée suivante,

— dans $U = \{u_1, \ldots, u_{\tilde{m}}\}$, l'indice $u \in \{0,1\}^{\tilde{m}}$ d'une clause, et

— dans $S = \{s_1, s_2\}$, la position $\gamma \in \{0,1\}^2$ d'un littéral dans la clause M_u,

le circuit C retourne le littéral $\mathrm{M}_{u,\gamma} = (\Lambda_w, \rho)$, avec

— dans $W = \{w_1, \ldots, w_{\tilde{n}}\}$, l'indice w de la variable Λ_w, et

— dans $\{p\}$, la polarité de ce littéral (0 si $\rho = \bot$ et 1 pour sinon).

On note $H = \{h_1, \ldots, h_\eta\}$ l'ensemble des éléments du circuit entre ses entrées et sorties. Le circuit C est acyclique. À part pour ses entrées, chaque élément de C calcule une des fonctions suivantes :

— une fonction constante (0 ou 1) depuis aucun voisin entrant, ou

— une fonction NOT depuis un voisin entrant, ou

— une fonction AND ou OR depuis deux voisins entrants, ou enfin

— seulement pour les sorties : une fonction identité depuis un voisin entrant.

Depuis C, la réduction consiste à construire le graphe G_C représenté dans la figure 3.6. L'idée est d'encoder le circuit C (qui lui-même encode une 3FNC Ψ) avec de simples contraintes sous la forme d'une 3FNC ψ. Cette méta-formule ψ portera sur n méta-variables $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ avec $n = \tilde{m} + \tilde{n} + \eta + 4$ et m méta-clauses $\mu = \{\mu_1, \ldots, \mu_m\}$ (à ne pas confondre avec les clauses $M_1, \ldots, M_{2^{\tilde{m}}}$ de la formule Ψ encodés par le circuit C) .

On va appeler de la même façon les composantes du circuit C et les sommets de l'ensemble $[n-1]$ du graphe G_ψ. En effet, on note $[n-1] = U \cup S \cup H \cup W \cup \{p\}$ et on pose $n = \nu$.

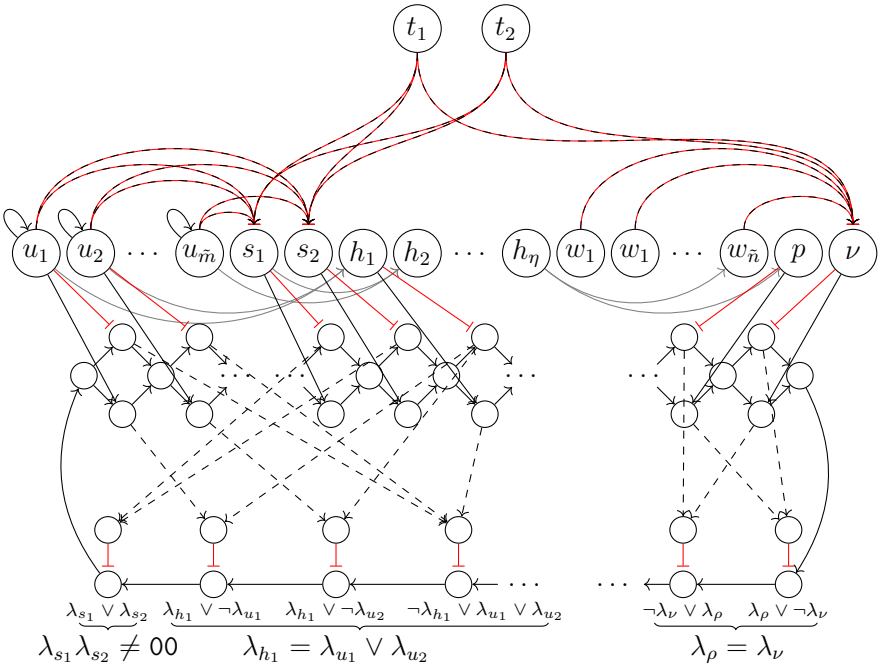


FIGURE 3.6 – Le graphe G_C encode l'instance SUCCINT-3SAT C, qui est lui-même un circuit encodant une formule Ψ (voir dans la figure 3.5). Les flèches en pointillés rouges et noires (en haut) représentent des arcs nuls. Nous avons $\Phi^{\max}(G) \geq 2^{\tilde{m}+1}$ si et seulement si Ψ a une solution

On va définir les clauses de la formule ψ de sorte que ses solutions (*i.e.* les mots $b \in \{0,1\}^n$ tel que $v^b(\psi) = \top$) respectent certaines contraintes. Pour donner une intuition, on veut que les fonctions locales sur les sommets de S, H et W correspondent aux fonctions du circuit C.

— On ajoute la clause $\lambda_{s_1} \vee \lambda_{s_2}$. Ceci nous assure que $b_S \neq 00$. En d'autres termes, b_S encode une position γ valide (*i.e.* entre 1 et 3).

— Pour chaque $h_i \in H$,

  — si h_i est une fonction constante 0 dans le circuit C alors on ajoute la clause $\neg\lambda_{h_i}$,

  — si h_i est une fonction constante 1 dans le circuit C alors on ajoute la clause λ_{h_i},

  — si h_i est une fonction NOT de h_j dans le circuit C alors on ajoute les deux clauses

$$(\lambda_{h_i} \vee \lambda_{h_j}) \wedge (\neg\lambda_{h_i} \vee \neg\lambda_{h_j}),$$

— si h_i est une fonction OR de h_j et h_k dans le circuit C alors on ajoute les trois clauses

$$(\lambda_{h_i} \vee \neg\lambda_{h_j}) \wedge (\lambda_{h_i} \vee \neg\lambda_{h_k}) \wedge (\neg\lambda_{h_i} \vee \lambda_{h_j} \vee \lambda_{h_k}),$$

— si h_i est une fonction AND de h_j et h_k dans le circuit C alors on ajoute les trois clauses

$$(\neg\lambda_{h_i} \vee \lambda_{h_j}) \wedge (\neg\lambda_{h_i} \vee \lambda_{h_k}) \wedge (\lambda_{h_i} \vee \neg\lambda_{h_j} \vee \neg\lambda_{h_k}).$$

On remarque que l'on peut « représenter » une clause à seulement 1 ou deux littéraux dans une 3FNC en lui faisant contenir plusieurs fois le même littéral. Considérons le numéro de clause u et la position γ encodée respectivement dans b_U et b_S. Les clauses ci-dessus nous assurent que pour chaque sommet $h_i \in H$, b_{h_i} a la même valeur que la composante h_i du circuit C quand en entrée il reçoit u et γ.

— Pour chaque $i \in W \cup \{p\}$ qui a comme voisin entrant h_j dans le circuit C, on ajoute les clauses

$$(\lambda_i \vee \neg\lambda_{h_j}) \wedge (\neg\lambda_i \vee \lambda_{h_j}).$$

Les composantes $W \cup \{p\}$ correspondent aux sorties du circuit. On s'assure avec ces clauses (et les clauses précédentes) que le numéro de variable w encodé dans b_W est égale à ce que l'on obtient en sortie du circuit quand en entrée il reçoit u et γ. On s'assure la même chose avec la polarité encodée dans p.

— On ajoute enfin les deux clauses

$$(\lambda_\nu \vee \neg\lambda_p) \wedge (\neg\lambda_\nu \vee \lambda_p).$$

Ces clauses nous assurent que $b_p = b_\nu$.

La formule ψ est maintenant définie, et donc G_ψ aussi.

Pour obtenir G_C depuis G_ψ, on ajoute deux sommets $T = \{t_1, t_2\}$ (sans aucun voisin entrant). On va aussi ajouter les arcs ci-dessous :

— pour tout $u_i \in U$, on ajoute une boucle positive sur u_i ;

— pour tout $t_i \in T$ et $j \in S \cup \{\nu\}$, on ajoute un arc nul de t_i vers j ;

— pour tout $u_i \in U$ et $s_j \in S$, on ajoute un arc nul de u_i vers s_j ;

— pour tout $w_i \in W$, on ajoute un arc nul de w_i vers ν ;

— pour tout $j \in H \cup W \cup \{p\}$, depuis chaque voisin entrant de j dans C vers j, on ajoute un arc étiqueté

  — 1 si j calcule la fonction identité, AND ou OR dans C, et

  — -1 si j calcule la fonction NOT dans C.

La construction de G_C est terminée. On a donc bien $G_C \in \mathcal{G}_\psi$.

Affirmation 3. *$\Phi^{\max}(G_C) = \alpha + 2^{\tilde{m}}$, avec $\alpha = \max_{\tilde{v}:\Lambda\to\{\bot,\top\}} |\{\mathrm{M}_j \in \mathrm{M} \mid \tilde{v}(\mathrm{M}_j) = \top\}|$.*

Démonstration. Dans un premier sens, prouvons que $\Phi^{\max}(G_C) \geq \alpha+m$. Considérons une valuation $\tilde{v} : \Lambda \to \{\bot, \top\}$ qui satisfait α clauses de Ψ. Prouvons l'existence du réseau $f \in \mathrm{F}(G_C)$ tel que $\phi(f) \geq \alpha + m$. Soit $I = V \setminus V_\psi$. Construisons $f_I \in \mathrm{F}_I(G)$ depuis v comme suit. Tout d'abord, on choisit $f_T : x \mapsto \mathtt{00}$. Notons que pour tout $x \in \{\mathtt{0},\mathtt{1}\}^{|V|}$, si $x_T = \mathtt{00}$ alors $f_T(x) = x_T$. En ce qui concerne f_ν, on choisit

$$f_\nu(x) = \begin{cases} x_{t_1} \oplus x_{t_2} \oplus \bigoplus_{i \in W} x_i & \text{si } x_T \neq \mathtt{00} \\ \mathtt{1} & \text{si } x_T = \mathtt{00} \text{ et } \tilde{v}(\Lambda_w) = \top \text{ avec } w = x_W, \\ 0 & \text{sinon.} \end{cases}$$

La première partie de la définition de f_ν nous assure que tous les arcs depuis $\mathcal{N}(\nu)$ vers ν sont effectifs. Les deux dernières parties nous assurent que la fonction f_ν encode bien $\tilde{v}$. En d'autres termes, pour tout $x \in \{\mathtt{0},\mathtt{1}\}^{|V|}$, si $x_T = \mathtt{00}$ alors $f_\nu(x) = \mathtt{1}$ si et seulement si $\tilde{v}(\Lambda_w) = \top$ avec $w = x_W$. On définit en utilisant la même astuce f_S, de manière à avoir la propriété suivante. On considère $x \in \{\mathtt{0},\mathtt{1}\}^{|V|}$ avec $x_T = \mathtt{00}$ et soit $u = x_U$. On veut que si $\tilde{v}(\mathrm{M}_u) = \top$, alors $f_S(x) = \gamma$ avec γ une position telle que $\mathrm{M}_{u,\gamma} = (\Lambda_i, \rho)$ et $\tilde{v}(\Lambda_i) = \rho$. Quand $\tilde{v}(\mathrm{M}_u) = \bot$, il n'y a pas de telle position γ et la valeur $f_S(x)$ n'importe pas (on peut alors poser $f_S(x) = \mathtt{01}$ par exemple). De plus, on définit $f_{u_i} : x \mapsto x_{u_i}$ pour tout $i \in [\tilde{m}]$ et chaque fonction f_j pour $j \in H \cup W \cup \{p\}$ calcule la même fonction que dans le circuit C.

D'après le lemme 6, on peut compléter la fonction f en gagnant la propriété suivante. Pour tout mot $z \in \{\mathtt{0},\mathtt{1}\}^{|V|}$ tel que $f_I(z) = z_I$ et $b = z_{[n]}$, alors $\phi_{z_I}(f) \geq \begin{cases} 2 & \text{si } v^b(\psi) = \top \\ 1 & \text{sinon} \end{cases}$. Soit $u \in \{\mathtt{0},\mathtt{1}\}^{\tilde{m}}$. On remarque que tous les cycles de G_C passant par l'ensemble de sommets de I sont des boucles sur les sommets de U et que $\tilde{f}_U(u) = u$. Donc, il existe un mot $x \in \{\mathtt{0},\mathtt{1}\}^I$ avec $x_U = u$ et tel que, pour tout $z \in \{\mathtt{0},\mathtt{1}\}^{|V|}$ avec $z_I = x$, on a $f_I(z) = z_I$. Il y a deux cas. Si $\tilde{v}(\mathrm{M}_u) = \bot$ alors on peut juste dire que $\phi_x(f) \geq 1$. Si $\tilde{v}(\mathrm{M}_u) = \top$ on peut montrer que $\phi_x(f) \geq 2$. En effet, prenons $b = z_{[n]}$. On voit qu'étant donné la façon dont on a défini f, chaque clause de μ est respectée. En effet :

— la valeur de b_S est une position γ valide ;

— chaque valeur b_j avec $j \in H \cup W \cup \{p\}$ est calculé exactement comme dans C ;

— enfin, posons $\gamma = b_S$, $w = b_W$ et ρ encodé dans b_p. Par définition de γ, on a $\mathrm{M}_{u,\gamma} = (\Lambda_W, \rho)$ et $\tilde{v}(\Lambda_W) = \rho$. De ce fait on a bien $b_p = b_\nu$.

Donc, $v^b(\psi) = \top$ et on a bien $\phi_x(f) \geq 2$.

On peut en conclure que f a deux (*resp.* un) points fixes pour chaque mot $u \in \{\mathtt{0},\mathtt{1}\}^{\tilde{m}}$ tel que la clause $\tilde{v}(\mathrm{M}_u) = \top$ (*resp.* $\bot$). Par conséquent, $\Phi^{\max}(G) \geq \phi(f) \geq \alpha + 2^{\tilde{m}}$.

Dans l'autre sens, considérons un réseau $f \in \mathrm{F}(G_C)$ tel que $\phi(f) = \Phi^{\max}(G_C)$. Montrons que $\phi(f) \leq \alpha + 2^{\tilde{m}}$. Construisons $\tilde{v} : \Lambda \to \{\bot, \top\}$ qui satisfait α clauses

de ψ. D'après le lemme 6, pour tout $x \in \{0,1\}^I$ et $b = x_{[n]} + \varepsilon^f$. On a $\phi_x(f) \leq \begin{cases} 2 & \text{si } v^b(\psi) = \top \\ 1 & \text{sinon} \end{cases}$. Remarquons que la fonction f_ν ne dépend que de T et W. De plus, f_T est une fonction constante qui a une unique image $y \in \{0,1\}^2$. Donc, tout point fixe $z \in \phi(f)$ respecte l'égalité $z_T = y$. Par conséquent, si deux points fixes $z, z' \in \phi(f)$ sont identiques sur W alors on a $z_\nu = z'_\nu = f_\nu(z)$. Si on se concentre sur les points fixes, on peut donc interpréter la fonction locale f_ν comme une fonction de valuation. En effet, elle prend l'indice $w \in \{0,1\}^{|W|}$ (d'une variable Λ_w) encodé dans z_W et sort une valeur booléenne qui correspond à une valeur logique $\bot$ ou $\top$. Plus formellement, construisons $\tilde{v} : \Lambda \to \{\bot, \top\}$ depuis f_ν comme suit. Considérons $w \in \{0,1\}^{\tilde{n}}$. Soit $w' = w + \varepsilon^f_W$, $a' = \tilde{f}_\nu(w'y)$ et $a = \varepsilon^f_\nu$. On pose $\tilde{v}(\lambda_w) = a$. Soit $u \in \{0,1\}^{|U|}$ et $u' = u + \varepsilon^f_U$. Remarquons que tous les cycles de G_C passant par les sommets I sont des boucles positives sur les sommets dans U. Donc, l'ensemble des configurations $z \in \{0,1\}^{|V|}$ telles que $z_U = u'$ sont identiques sur toutes les coordonnées de I. Notons $x = z_I$, $b' = x_{[n]}$ et $b = b' + \varepsilon^f$. On va montrer que si $\phi_x(f) \geq 2$ alors $\tilde{v}(\Lambda_u) = \top$. Supposons que $\phi_x(f) \geq 2$. On peut poser $\gamma = b_S$, $w = b_W$, ρ la valeur encodée x_p et, bien sûr, on a $u = b_U$. Comme $\phi_x(f) \geq 2$, toutes les clauses de μ sont respectées. Et donc $\mathrm{M}_{u,\gamma} = (\Lambda_w, \rho)$ et $b_p = b_\nu$. Par définition de $\tilde{v}$, cela veut dire que $\tilde{v}(\lambda_w) = \rho$ et donc que $\tilde{v}(\mathrm{M}_u) = \top$.

Donc, pour chaque clause M_u telle que $\tilde{v}(\mathrm{M}_u) = \top$, f a au plus deux points fixes. Pour les autres, il en a au plus un. Donc, $\Phi^{\max}(G_C) = \phi(f) \leq \alpha + 2^{\tilde{m}}$.

■

Finalement on a $\Phi^{\max}(G_C) = \alpha + 2^{\tilde{m}}$. En posant $k = 2^{\tilde{m}+1}$, on a $\Phi^{\max}(G_C) \geq k$ si et seulement si Ψ a une solution. Ceci conclut notre réduction. Le problème est donc bien NEXPTIME-difficile et NEXPTIME-complet.

□

3.6 Minimum Fixed Point Problem

Dans cette section, nous allons étudier les problèmes liés au nombre minimum de points fixes. La classe $\mathsf{NP}^{\mathsf{NP}}$ (souvent notée $\sum_2^P$) correspond aux problèmes calculables par une machine de Turing non déterministe avec un oracle dans NP. Le problème de décision suivant est complet pour la classe $\mathsf{NP}^{\mathsf{NP}}$ (voir le théorème 17.10 du Livre [116]).

QUANTIFIED SATISFIABILITY avec 2 alternances de quantifiers (QSAT_2)
Entrée: Une 3FNC ψ sur $\{\lambda_1, \ldots, \lambda_n\}$ et un entier $s \in [n]$.
Sortie: $\exists v : \{\lambda_1, \ldots, \lambda_s\} \to \{\bot, \top\}$, $\forall v' : \{\lambda_{s+1}, \ldots, \lambda_n\} \to \{\bot, \top\}$, $v \cup v'(\psi) = \top$?

Théorème 10. *Les problèmes* MinFPP *et* k-MinFPP *pour tout* $k \geq 1$ *sont* NEXPTIME*-complet. Quand on se restreint aux graphes dont le degré est borné par une constante,* k-MinFPP *devient* $\mathsf{NP}^{\mathsf{NP}}$*-complet pour tout* $k \geq 1$ *et* MinFPP *devient* $\mathsf{NP}^{\#\mathsf{P}}$*-complet.*

Le théorème 10 est une conséquence directe des lemmes 10, 13 et 14.

Le lemme 10 ci-dessous prouve que les inclusions données dans le théorème 10 sont vraies.

Lemme 10. *Le problème* MinFPP *est dans* NEXPTIME*. Quand on se restreint aux graphes dont le degré est borné par une constante,* k-MinFPP *est dans* $\mathsf{NP}^{\mathsf{NP}}$ *pour tout* $k \geq 1$ *et* MinFPP *est dans* $\mathsf{NP}^{\#\mathsf{P}}$*.*

Démonstration. On a déjà expliqué dans la section 3.2 pourquoi le problème MinFPP (comme tous les problèmes de ce chapitre) était dans NEXPTIME.

On va donc considérer que le degré est borné.

Pour résoudre le problème k-MinFPP avec $k \geq 1$, on peut exécuter l'algorithme suivant.

— On devine la fonction $f \in \mathrm{F}(G)$ (il en existe un nombre linéaire).

— On crée la formule ψ qui correspond à $f(x) = x$ (*i.e.* $f_1(x) = x_1 \wedge \ldots f_n(x) = x_n$).

— On pose à l'oracle le problème « est-ce que ψ a plus de k solutions ? ». C'est bien un problème dans NP car si la réponse est oui alors l'oracle peut le justifier avec un certificat polynomial : les k solutions.

— Si la réponse est oui alors $\phi(f) \geq k$ et on répond non. Dans le cas contraire $\phi(f) < k$ et on répond oui.

On accepte si une branche répond oui. Le problème est bien dans $\mathsf{NP}^{\mathsf{NP}}$.

L'algorithme pour MinFPP est quasiment identique à celui de MaxFPP.

— On devine la fonction $f \in \mathrm{F}(G)$ (il en existe un nombre linéaire).

— On crée la formule ψ qui correspond à $f(x) = x$ (*i.e.* $f_1(x) = x_1 \wedge \ldots f_n(x) = x_n$).

— On pose à l'oracle #P le problème « combien ψ a de solutions ? ».

— La réponse nous donne $\phi(f)$ et il suffit de répondre oui si et seulement si $\phi(f) < k$.

On accepte si une branche répond oui. Le problème est bien dans $\mathsf{NP}^{\#\mathsf{P}}$. □

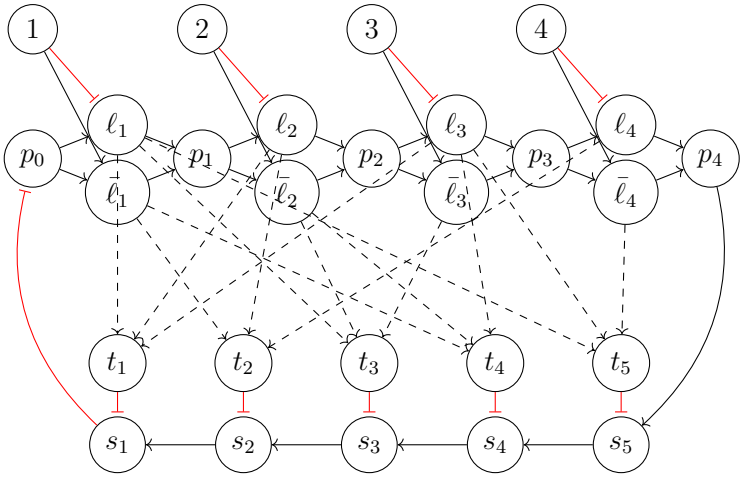


Figure 3.7 – GIS G^-_ψ présenté dans la définition 3. La 3FNC ψ considérée représente la formule suivante :
$(\lambda_1 \vee \lambda_2 \vee \lambda_3) \wedge (\neg\lambda_1 \vee \lambda_2 \vee \lambda_4) \wedge (\lambda_1 \vee \neg\lambda_2 \vee \neg\lambda_3) \wedge (\neg\lambda_1 \vee \neg\lambda_2 \vee \lambda_3) \wedge (\lambda_1 \vee \lambda_3 \vee \neg\lambda_4$
La 3FNC ψ a une solution si et seulement si $\Phi^{\min}(G_\psi) \leq 0$. Sinon $\Phi^{\min}(G_\psi) = 1$.

Pour prouver la difficulté des problèmes qui nous intéressent dans cette section, nous allons d'abord définir le GIS G^-_ψ (illustré dans la figure 3.7).

Définition 3 (G^-_ψ et $\mathcal{G}^-_\psi$). *On définit $G^-_\psi = (V', A', \sigma')$ exactement comme G_ψ mais avec $\sigma'_{s_1,p_0} = -1$. On définit $\mathcal{G}^-_\psi$ comme on a défini $\mathcal{G}_\psi$ mais en remplaçant G_ψ par G^-_ψ.*

Pour prouver la difficulté des problèmes qui nous intéressent dans cette section, on va introduire 2 lemmes. Le lemme 11 ci-dessous est l'équivalent du lemme 5 mais avec $\mathcal{G}^-_\psi$ en lieu et place de $\mathcal{G}_\psi$.

Lemme 11. *Soit ψ, une 3FNC sur $\lambda = \{\lambda_1, \ldots, \lambda_n\}$, $G = (V, A, \sigma) \in \mathcal{G}^-_\psi$, et $I = V \setminus \tilde{V}_\psi$. Il existe une fonction $f_{\tilde{V}_\psi} \in \mathrm{F}_{\tilde{V}_\psi}(G)$ telle que pour tout $f_I \in \mathrm{F}_I(G)$, pour tout $z \in \{0,1\}^V$ et pour $b = z_{[n]}$, si $f_I(z) = z_I$, alors le réseau f (défini en « concaténant » f_I et $f_{\tilde{V}_\psi}$) respecte*

$$\phi_{z_I}(f) \leq \begin{cases} 0 & \text{si } v^b(\psi) = \top \\ 1 & \text{sinon} \end{cases}.$$

Démonstration. Soit $\tilde{V} = \tilde{V}_\psi$. On va définir les fonctions locales f_i avec $i \in \tilde{V}$ de la même façon que dans le lemme 5. En d'autres termes, pour tout $i \in L \cup \bar{L} \cup S \cup T$, f_i est la fonction AND et pour tout $i \in P$, f_i est la fonction OR.

Notons que contrairement au lemme 5, l'arc (s_1, p_0) est ici négatif. La seule fonction $f_{p_0} \in \mathrm{F}_{p_0}(G)$ possible est donc $f_{p_0} : x \mapsto \neg x_{s_1}$.

Montrons tout d'abord que $\phi_{z_I}(f) \leq 1$. On remarque que tous les cycles de G dans $\tilde{V}$ passent par p_0. D'après le corolaire 2, si $\phi_{z_I}(f) \geq 2$, alors il existe deux configurations $x, y \in \Phi_{z_i}(f)$ telles que $0 = x_{p_0} < y_{p_0} = 1$.

Montrons qu'une telle configuration x (*i.e.* telle que $x_{p_0} = 0$) ne peut pas être un point fixe. Supposons par l'absurde qu'il existe bien $x \in \Phi_{z_I}(f)$ avec $x_{p_0} = 0$. On peut montrer par récurrence que pour tout $i \in [0, n]$, $x_{p_i} = 0$. En effet, c'est vrai pour $i = 0$. Supposons ensuite que ce soit vrai pour $i - 1 \in [0, n-1]$. On a $x_{\ell_i} = x_{\bar{\ell}_i} = x_{p_{i-1}} \wedge \cdots = 0 \wedge \cdots = 0$ et $x_{p_i} = x_{\ell_i} \vee x_{\bar{\ell}_i} = 0 \vee 0 = 0$. L'hypothèse de récurrence est donc vraie pour tout $i \in [n]$ et donc, en particulier, $x_{p_n} = 0$. De la même façon, on peut prouver par récurrence descendante que pour tout $j \in [m+1]$, $x_{s_j} = 0$. On rappelle que l'on note $s_{m+1} = x_{p_n}$. C'est donc vrai pour $j = m + 1$. Supposons maintenant que c'est vrai pour $j + 1 \in [2, m+1]$. On a $x_{s_j} = x_{s_{j+1}} \wedge \neg x_{t_j} = 0 \wedge \neg x_{t_j} = 0$. L'hypothèse de récurrence est donc vraie pour tout $j \in [m+1]$ et donc, en particulier, $x_{s_1} = 0$. Mais on a $f_{p_0}(x) = \neg x_{s_1} = 1 \neq 0 = x_{p_0}$. La configuration x n'est donc pas un point fixe, ce qui est une contradiction. Donc, $\phi_{z_I}(f) \leq 1$.

Maintenant, montrons que si $v^b(\psi) = \top$ alors $\phi_{z_I}(f) = 0$. Supposons que $v^b(\psi) = \top$. On a déjà vu qu'il n'existe pas $x \in \Phi_{z_I}(f)$ avec $x_{p_0} = 0$. Supposons par l'absurde qu'il existe $y \in \Phi_{z_I}(f)$ avec $y_{p_0} = 1$. On peut prouver par récurrence, que pour tout $i \in [0, n]$, $y_{p_i} = 1$. C'est vrai pour $i = 0$. Supposons que c'est vrai pour $i - 1 \in [0, n-1]$. Il y a deux cas. Si $y_i = 0$ alors on a $y_{\ell_i} = y_{p_{i-1}} \wedge \neg b_i = 1 \wedge 1 = 1$. Si $y_i = 1$ alors on a $y_{\bar{\ell}_i} = y_{p_{i-1}} \wedge b_i = 1 \wedge 1 = 1$. Dans les deux cas, on a $y_{p_i} = y_{\ell_i} \vee y_{\bar{\ell}_i} = 1$. L'hypothèse de récurrence est donc vraie pour tout $i \in [n]$ et donc, en particulier, $y_{p_n} = 1$. En outre, on remarque que pour tout $i \in [n]$, $y_{\ell_i} = 0$ si et seulement si $b_i = 1$. De plus, $y_{\bar{\ell}_i} = 0$ si et seulement si $b_i = 0$. On peut montrer que pour tout $j \in [m]$, $y_{t_j} = 0$. En effet, on a supposé que $v^b(\psi) = \top$. Donc, pour toute clause μ_j, il existe un littéral $\mu_{j,\gamma} = (\lambda_i, \rho)$ tel que $v^b(\lambda_i) = \rho$. Il y a deux cas :

— si $\rho = \top$ alors $b_i = 1$ et $y_{\ell_i} = 0$. De plus, $(\ell_i, t_j) \in A$. On a donc $y_{t_j} = y_{\ell_i} \wedge \cdots = 0 \wedge \cdots = 0$;

— si $\rho = \bot$ alors $b_i = 0$ et $y_{\bar{\ell}_i} = 0$. De plus, $(\bar{\ell}_i, t_j) \in A$. On a donc $y_{t_j} = y_{\bar{\ell}_i} \wedge \cdots = 0 \wedge \cdots = 0$.

Donc, on a bien $y_{t_j} = 0$ pour tout $j \in [m]$. Pour finir, on peut montrer par récurrence que pour tout $j \in [m+1]$, on a $y_{s_j} = 1$. C'est vrai pour $j = m + 1$. Maintenant, supposons que c'est vrai pour $j + 1 \in [2, m+1]$. On a $y_{s_j} = \neg y_{t_j} \wedge y_{s_{j+1}} = \neg 0 \wedge 1 = 1$. L'hypothèse de récurrence est donc vraie pour tout $j \in [m+1]$ et donc, en particulier, $y_{s_1} = 1$. Mais $f_{p_0}(y) = \neg y_{s_1} = 1 \neq 0 = y_{p_0}$. Donc, y n'est pas un point fixe, et $\phi_{z_I}(f) = 0$.

□

Le lemme 12 ci-dessous est l'équivalent du lemme 6 mais avec $\mathcal{G}_\psi^-$ à la place de $\mathcal{G}_\psi$.

Lemme 12. *Soit ψ, une* 3FNC *sur $\lambda = \{\lambda_1, \ldots, \lambda_n\}$, $G = (V, A, \sigma) \in \mathcal{G}_\psi$ et $f \in \mathrm{F}(G)$. Soit $z \in \{0,1\}^{|V \setminus \tilde{V}_\psi|}$ et $b = z_{[n]} + \varepsilon^f$. Si $v^b(\psi) = \bot$, alors $\phi_z(f) \geq 1$.*

Démonstration. Soit $\tilde{V} = \tilde{V}_\psi$. On va supposer que $\phi_z(f) = 0$ et on va montrer que cela implique que $v^b(\psi) = \top$. On définit x et $y \in \{0,1\}^{|V|}$ comme suit :

— $x_{V\setminus\tilde{V}} = y_{V\setminus\tilde{V}} = z$;

— $x_{p_0} < y_{p_0}$;

— $x_{\tilde{V}\setminus\{p_0\}} = f_{\tilde{V}\setminus\{p_0\}}(x)$ et $y_{\tilde{V}\setminus\{p_0\}} = f_{\tilde{V}\setminus\{p_0\}}(y)$.

On remarque que sans p_0, le graphe G serait acyclique sur $\tilde{V}$. Donc, il y a seulement un couple x, y qui correspond à cette définition. Pour $* \in \{<, >, \leq, \geq, =\}$, soit $I_* = \{i \in V \mid x_i * y_i\}$. On peut voir que pour tout v, si $\mathcal{N}(v) \subseteq I_\leq$ et que pour tout $v' \in \mathcal{N}(v)$, on a $\sigma_{v',v} = 1$ alors $v \in I_\leq$. En utilisant cette propriété on peut facilement prouver par récurrence que $P \cup L \cup \bar{L} \subseteq I_\leq$ et donc que $T \subseteq I_\leq$.

Maintenant, on sait que $f_{p_0}(x) \neq x_{p_0}$ et $f_{p_0}(y) \neq y_{p_0}$ sinon x ou y appartient à $\Phi_z(f)$ et $\phi_z(f) \neq 0$. Par conséquent $0 = x_{p_0} \neq f_{p_0}(x) = \neg x_{s_1}$ et donc $x_{s_1} = 0$. De plus, $1 = y_{p_0} \neq f_{p_0}(y) = \neg y_{s_1}$ et donc $y_{s_1} = 1$. Cela veut dire que $s_1 \in I_<$.

On peut maintenant prouver par récurrence que pour tout $j \in [m+1]$, $s_j \in I_<$. D'abord, c'est vrai pour $j = 1$. Ensuite, supposons que c'est vrai pour $j - 1 \in [m]$. D'abord, prouvons que $t_{j-1} \in I_=$. On sait déjà que $t_{j-1} \in I_\leq$. Supposons par l'absurde que $t_{j-1} \in I_<$. Il y a deux cas :

— si $f_{s_{j-1}}$ est la fonction OR alors $x_{s_{j-1}} = \neg x_{t_{j-1}} \vee x_{s_j} = 1 \vee \cdots = 1$;

— si $f_{s_{j-1}}$ est la fonction AND alors $y_{s_{j-1}} = \neg y_{t_{j-1}} \wedge y_{s_j} = 0 \wedge \cdots = 0$.

Dans les deux cas, $s_{j-1} \notin I_<$, ce qui contredit l'hypothèse de récurrence. Donc, $t_{j-1} \in I_=$. Rappelons que $\mathcal{N}(s_{j-1}) = \{t_{j-1}, s_j\}$, que $s_{j-1} \in I_<$ et que l'arc (s_j, s_{j-1}) est positif. On a donc clairement, $s_j \in I_<$ et par récurrence $S \subseteq I_<$. En outre, on a prouvé que $T \subseteq I_=$.

Avec un raisonnement similaire, on peut prouver par récurrence descendante que pour tout $i \in [n]$, $\ell_i \in I_<$ ou $\bar{\ell}_i \in I_<$ et $p_i \in I_<$. Il y a donc un cycle C négatif qui contient P, S, et pour tout $i \in [n]$, ℓ_i ou $\bar{\ell}_i$. De plus, on peut prendre le cycle C tel que pour tout $c \in C$, c appartient à $I_<$. On va considérer que si C peut contenir ℓ_i ou $\bar{\ell}_i$ au choix, alors il contient ℓ_i. Maintenant, avec exactement la même démonstration que dans l'Affirmation 1, on prouve que pour tout $i \in [n]$, on a $\ell_i \in C \iff b_i = 0$. On sait que pour tout $j \in [m]$, $t_j \in I_=$. On rappelle que tous les arcs entrants de t_j sont positifs et que $L \cup \bar{L} \subseteq I_\leq$. Donc, clairement il existe un sommet $\ell = \ell_i$ ou $\ell = \bar{\ell}_i$ tel que $\ell \in \mathcal{N}(t_j)$ et $\ell \in I_=$. Comme $\ell \in I_=$, on sait que $\ell \notin C$. Ce voisin entrant ℓ correspond à un littéral $\mu_{j,\gamma} = (\lambda_i, \rho)$. Il y a deux possibilités.

— Si $\rho = \top$, alors $\ell = \ell_i \in L$. De plus, $\ell_i \notin C$ car $\ell \in I_=$. Donc, $v^b(\lambda_i) = \top = \rho$ et $v^b(\mu_j) = \top$.

— Si $\rho = \bot$, alors $\ell = \bar{\ell}_i \in \bar{L}$. De plus, $\bar{\ell}_i \notin C$ car $\bar{\ell}_i \in I_<$, et donc $\ell_i \in C$. Donc, $v^b(\lambda_i) = \bot = \rho$ et $v^b(\mu_j) = \top$.

Pour toute clause, μ_j, on a $v^b(\mu_j) = \top$. Donc, si $\phi_z(f) = 0$ alors v^b est une solution de ψ.

□

Lemme 13. *Si* $\Delta(G) \leq d$*,* k-MinFPP *est* $\mathsf{NP}^{\mathsf{NP}}$*-difficile pour tout* $k \geq 1$ *et* MinFPP *est* $\mathsf{NP}^{\#\mathsf{P}}$*-difficile.*

Démonstration. On définit le graphe $G^-_{\psi,s} = (V, A, \sigma)$ qui est similaire au graphe $G_{\psi,s}$ mais avec $\sigma_{s_1,p_0} = -1$. Clairement, $G^-_{\psi,s} \in \mathcal{G}^-_\psi$. On remarque que l'on peut utiliser la même astuce que dans la remarque 3 appliquée au graphe $G^-_{\psi,s}$ pour limiter $\Delta(G^-_{\psi,s})$ à 2. On peut utiliser le même raisonnement que dans le lemme 9, en utilisant les lemmes 11 et 12 à la place des lemmes 5 et 6. En faisant cela, on conclut que

$$\Phi^{\min}(G^-_{\psi,s}) = \min_{v:\{\lambda_1,\ldots,\lambda_s\}\to\{\bot,\top\}} 2^{n-s} - \alpha(v),$$

avec $\alpha(v) = |\{v' : \{\lambda_{s+1}, \ldots, \lambda_n\} \to \{\bot, \top\} \mid v \cup v'(\psi) = \top\}|$.

Maintenant posons $k = 2^n/2$. On voit que le résultat de E-Maj3SAT(ψ, s) est équivalent au résultat de MinFPP$(G^-_{\psi,s}, k)$. Donc, MinFPP est $\mathsf{NP}^{\#\mathsf{P}}$-difficile.

On voit aussi que $\Phi^{\min}(G^-_{\psi,s}) = 0$ correspond au cas où il existe une valuation $v : \{\lambda_1, \ldots, \lambda_s\} \to \{\bot, \top\}$ telle que pour tout $v' : \{\lambda_{s+1}, \ldots, \lambda_n\} \to \{\bot, \top\}$, on a $v \cup v'(\psi) = \top$. On voit donc que $\mathrm{QSAT}_2(\psi, s)$ est équivalent à 1-MinFPP(ψ). Donc, 1-MinFPP est $\mathsf{NP}^{\mathsf{NP}}$-difficile.

Pour $k \geq 2$, on peut prendre $\tilde{k} = \lceil \log_2(k) \rceil$. On a donc $k \leq 2^{\tilde{k}}$. Soit G' qui correspond à G auquel on ajoute $\tilde{k}$ sommets isolés avec une boucle positive sur chacun d'eux. Clairement $\Phi^{\min}(G') = 0$ si $\mathrm{QSAT}_2(\psi, s)$ et $\Phi^{\min}(G') \geq 2^{\tilde{k}} \geq k$ sinon. Autrement dit k-MinFPP(G') équivaut à $\mathrm{QSAT}_2(\psi, s)$. En conclusion, pour tout $k \geq 1$, k-MinFPP est $\mathsf{NP}^{\mathsf{NP}}$-difficile.

□

Lemme 14. MinFPP *en général est* $\mathsf{NEXPTIME}$*-difficile.*

Démonstration. On considère un circuit C similaire à celui décrit dans le théorème 9. On définit le graphe $G^-_C = (V, A, \sigma)$ qui est similaire au graphe G_C mais avec $\sigma_{s_1,p_0} = -1$. On peut utiliser le même raisonnement que dans le lemme 9, en utilisant les lemmes 11 et 12 à la place des lemmes 5 et 6. En faisant cela, on conclut que $\Phi^{\min}(G^-_C) = 0$ si et seulement si la formule ψ décrite par le circuit C a une solution. Autrement dit 1-MinFPP(G^-_C) est équivalent à Succint-3SAT(C). Donc, le problème 1-MinFPP est $\mathsf{NEXPTIME}$-difficile. D'une manière générale, pour tout $k \geq 1$, 1-MinFPP est $\mathsf{NEXPTIME}$-difficile. En effet, il suffit d'utiliser l'astuce qui consiste à ajouter des sommets isolés avec une boucle positive sur chacun d'eux. À plus forte raison, MinFPP est $\mathsf{NEXPTIME}$-difficile.

□

3.7 Conclusion et perspectives

Dans ce chapitre, on a étudié la complexité algorithmique de plusieurs problèmes de décision qui paraissent plutôt naturels. À l'exception du problème 1-MAXFPP, on connait exactement la classe de complexité de chacun de ces problèmes. On a d'ailleurs rencontré des classes de complexité diversifiées, dont certaines sur lesquelles la littérature est plutôt réduite.

Bien qu'il s'agisse de problèmes naturels, ils n'avaient jamais été étudiés avant. En effet, la plupart des travaux sur la complexité des réseaux d'automates booléens sont de la forme suivante. Étant donné un réseau f, on veut savoir s'il respecte une certaine propriété dynamique P. Ici, on considère plutôt un graphe d'interaction G et on veut savoir si au moins un réseau $f \in \mathrm{F}(G)$ respecte la propriété P. Dans certains cas, ce nouveau problème est beaucoup plus facile. Par exemple, le problème de savoir si un RAB (codé en forme normale conjonctive par exemple) a un point fixe est NP-complet en général [97] alors que l'on a montré que trouver un réseau dans $\mathrm{F}(G)$ est un problème polynomial. À l'inverse, vérifier qu'un réseau f n'a aucun point fixe est un problème dans coNP alors que même si G a un degré borné, vérifier qu'il existe un réseau $f \in \mathrm{F}(G)$ sans point fixe est déjà dans $\mathsf{NP}^{\mathsf{NP}}$.

Il reste beaucoup de travail à réaliser sur ce thème. D'abord, on a prouvé que 1-MAXFPP est équivalent au problème de trouver un cycle pair dans un digraphe. On sait que ce problème est dans P [105, 135] mais à notre connaissance aucun travail n'a été fait pour montrer sa C-difficulté pour une quelconque classe C. Trouver des bornes inférieures serait intéressant, car le problème est équivalent à de nombreux autres problèmes de théorie des graphes dont le résumé de l'article [135] donne déjà une liste importante.

De plus, on pourrait étudier l'effet d'autres restrictions sur des graphes. Par exemple, que se passe-t-il quand on se restreint aux graphes d'interaction composés uniquement d'arcs positifs ? C'est un problème intéressant, car les réseaux admis par ces digraphes sont des réseaux monotones qui, depuis les travaux de Tarski [146], ont été très étudiés [4, 8, 15, 84, 85, 86, 107, 134]. On pense que l'on pourrait adapter notre construction pour qu'elle fonctionne avec cette contrainte supplémentaire. Nous pensons que dans ce cas-là, il existe un entier k_0 tel que le problème k-MAXFPP est dans P pour tout $k \leq k_0$ et dans NP sinon.

Comme dans le prochain chapitre, on pourrait étudier les graphes non signés par exemple, ou sur des alphabets non booléens (comme dans le prochain chapitre). Autre possibilité : on pourrait s'autoriser à « ignorer » certains arcs du graphe d'interaction. Ceci pourrait changer énormément la difficulté du problème (a priori, en les rendant plus facile). Ce problème se rapprocherait alors plus des problèmes de *network coding* et de théorie de l'information [69, 101].

Finalement, la famille des problèmes similaires à celui introduit dans ce chapitre (où on fixe G et une propriété P) est quasiment illimitée car il y a énormément de propriétés dynamiques intéressantes. Les autres propriétés dynamiques qui viennent à l'esprit quand on parle de réseaux d'automates sont surement la taille et le nombre de cycles limites et la taille des trajectoires transitoires. On pourrait aussi se pencher sur des propriétés sur le bassin d'attraction, des résultats de complexité là-dessus étant déjà connus sur les réseaux booléens à seuil [59].

Chapitre 4

Expansivité

4.1 Introduction

Dans ce chapitre, on introduit le concept de réseaux d'automates expansifs. De manière intuitive, on dit qu'un réseau d'automates est expansif si on a la propriété suivante : on peut déterminer sa configuration initiale simplement en observant l'évolution d'une de ses composantes pendant un temps fini. Plus formellement, $f \in \mathrm{F}(n, q)$ est expansif s'il satisfait les deux conditions équivalentes suivantes :

1. Pour toute coordonnée $i \in [n]$, il existe un entier s tel que la trace $\rho_i^{(s)} : \mathbb{A}^n \to \mathbb{A}^s$ définie par $\rho^{(s)}(x) := (f_i(x), f_i^2(x), \dots, f_i^s(x))$ est injective [1].
2. Pour toute coordonnée $i \in [n]$ et pour toute paire de configurations $\{x, y\} \subseteq \mathbb{A}^n$, il existe un entier $t \geq 1$ tel que $f_i^t(x) \neq f_i^t(y)$.

Nous sommes surtout intéressés par des résultats généraux sur les réseaux d'automates, sans aucune application particulière à l'esprit. Néanmoins, les réseaux d'automates sont polyvalents et peuvent être vus et utilisés de différentes manières. Ainsi, le concept de réseaux d'automates expansifs introduit ici est significatif dans plusieurs domaines.

Système dynamique. Le mot « expansif » est classiquement utilisé en théorie des systèmes dynamiques et correspond à une forte imprévisibilité topologique [88]. Dans le cas des automates cellulaires [98], cette notion topologique a une interprétation concrète en termes de traces : l'orbite d'un système peut être déduite de sa trace temporelle sur une région spatiale limitée (il existe une notion similaire en dynamique symbolique [23]). La définition ci-dessus suit la même idée et il y a en fait une correspondance précise entre l'expansivité des automates cellulaires et celle des réseaux d'automates (voir la section 4.9).

Calcul distribué. Un réseau d'automate expansif peut être vu comme un protocole qui résout le problème de donner la configuration globale du réseau à chacune de ses composantes. En outre si la constante s dans la première partie de la définition ci-dessus est optimale (*i.e.* égale à n), alors le réseau a une autre propriété algorithmique intéressante : il existe une bijection entre les configurations initiales du réseau ($x \in \mathbb{A}^n$) et les n premiers états d'un de ses automates ($f_i(x), \dots, f_i^n(x)$). De tels réseaux expansifs existent comme on va le montrer.

Outil de modélisation et sciences expérimentales. Avec ce point de vue général à l'esprit, on peut s'intéresser à la prédiction d'un système à partir d'observations partielles. En particulier, certaines composantes peuvent être difficiles ou impossibles à observer. Les réseaux d'automates expansifs correspondent à un cas favorable où observer n'importe quelle composante un certain temps est suffisant. Il peut aussi être intéressant de généraliser le type d'observations faites sur le système. C'est ce que l'on fait dans la section 4.7 où on présente une plus forte forme d'expansivité.

1. La fonction f^i est la composition de i fois la fonction f et ne doit pas être confondue avec l'instruction $f^{(i)}$. En outre, $f_i^s(x)$ est juste une manière concise d'écrire $(f^s(x))_i$.

Tableaux orthogonaux et codes parfaits Dans un réseau expansif, l'orbite d'une configuration contient beaucoup d'informations redondantes car connaitre s états consécutifs de n'importe quelle composante suffit à reconstruire l'orbite complète. En poussant l'idée plus loin, on obtient une forme plus forte d'expansivité dans la section 4.7. Cette forme donne des tableaux orthogonaux d'indice 1 ou de manière équivalente des codes parfaits (voir [79, 103]). Plus précisément l'ensemble des orbites d'une certaine taille forment un code, et, dans le cas linéaire, cela peut être représenté de manière compacte en donnant la fonction f qui définit le réseau.

Présentons maintenant le plan de ce chapitre. D'abord, dans la section 4.3, on étudie les graphes d'interaction qui admettent des réseaux d'automates expansifs. Le théorème 13 affirme que ce sont exactement les digraphes fortement connexes et *couvrables* (*i.e.* tels qu'il existe un ensemble de cycles disjoints qui contiennent tous les sommets du graphe). Le corolaire 5 montre que, pour un tel digraphe G, presque tous les réseaux linéaires de $\mathrm{F}(G, q)$ sont expansifs avec q la cardinalité d'un corps fini suffisamment large.

Ensuite, dans la section 4.4, on présente deux classes de graphes d'interaction qui admettent toujours un réseau expansif pour tout alphabet. La première est la famille des cycles bouclés (voir la proposition 4) et la seconde est la famille des cycles de cycles (voir la proposition 5).

À l'inverse, dans la section 4.5, on se concentre sur des graphes d'interaction qui n'admettent aucun réseau expansif pour un alphabet donné. Le théorème 15 montre que, pour tout alphabet de taille q, il existe un graphe d'interaction G fortement connexe et couvrable tel que $\mathrm{F}(G, q)$ ne contient aucun réseau expansif. On présente également un graphe qui admet un réseau expansif pour tout alphabet mais pas d'expansif linéaire pour une infinité de tailles d'alphabets (voir la proposition 6).

Dans la section 4.6, on s'intéresse au *temps d'expansivité* $\iota(f)$ d'un réseau expansif f. Il s'agit du nombre maximum d'étapes de temps nécessaires pour calculer n'importe quelle configuration initiale depuis chaque coordonnée. C'est également la plus petite valeur de s de la définition d'expansivité donnée plus tôt. On prouve que le temps d'expansivité de $f \in \mathrm{F}(n, q)$ peut varier entre n et presque q^n (voir le théorème 17). On prouve aussi que le temps minimum n est obtenu par un réseau linéaire quand q est une puissance d'un nombre premier.

De plus, dans la section 4.7, on considère une notion plus forte d'expansivité. On veut pouvoir retrouver la configuration initiale à partir de n'importe quelle observation suffisamment grande du système (pas forcément sur une seule coordonnée). Nous montrons qu'un réseau d'automates avec cette propriété donne des *codes parfaits* (voir la proposition 9). De plus, on prouve qu'il existe un tel réseau si et seulement si le graphe d'interaction est complet (voir le théorème 18), mais qu'il requiert un alphabet de taille quadratique par rapport à la taille du graphe (voir le corolaire 7).

Enfin, dans la section 4.8, on étudie le nombre moyen de différences entre deux orbites. On appelle ce nombre la *fréquence d'expansivité*. On montre que ce nombre peut être arbitrairement proche de 1 (voir le théorème 19). La section 4.6 contient une construction qui montre qu'elle peut également être arbitrairement proche de 0.

En conclusion, dans la section 4.9, on discute de la relation entre notre définition d'expansivité pour les réseaux d'automates, et celle donnée pour les automates cellulaires.

Les résultats présentés dans ce chapitre ont fait l'objet d'un papier qui est en cour de relecture en conférence [26].

4.2 Définitions et résultats préliminaires

Nous rappelons dans les sous-sections 4.2.1 et 4.2.1 quelques résultats classiques d'algèbre [99] et présentons les notations que nous allons utiliser par la suite. Nous présentons des résultats moins classiques dans la sous-section 4.3, et des résultats préliminaires dans les sous-sections 4.2.3 et 4.2.4.

4.2.1 Groupes et corps finis

Un *groupe* $\mathcal{G}$ est composé d'un ensemble E et d'une *loi de composition* $\bigcirc$. La loi de composition $\bigcirc$ est une application de E^2 dans E. L'image de $(a, b) \in E^2$ par $\bigcirc$ se note $a \bigcirc b$. De plus, la loi $\bigcirc$ respecte trois propriétés :

— elle est *associative* : pour tout $a, b, c \in E$, $(a \bigcirc b) \bigcirc c = a \bigcirc (b \bigcirc c)$;

— elle possède un *élément neutre* : il existe un élément $e \in E$ tel que, pour tout $a \in E$, $a \bigcirc e = e \bigcirc a = a$;

— elle est *symétrique* : pour tout $a \in E$, il existe $b \in E$ tel que $a \bigcirc b = b \bigcirc a = e$.

Dans ce chapitre, on ne va travailler qu'avec des groupes finis. Plus précisément, l'ensemble E de définition de nos groupes sera souvent alphabet $\mathbb{A} = [0, q[$.

Un *morphisme* entre un groupe $\mathcal{G} = (E, \bigcirc)$ et un autre groupe $\mathcal{G}' = (E', \square)$ est une application $m : E \to E'$ qui respecte $m(a \bigcirc b) = m(a) \square m(b)$. Un *endomorphisme* e du groupe $\mathcal{G} = (E, \bigcirc)$ est un morphisme de $\mathcal{G}$ dans lui-même. Les endomorphismes de groupes ont certaines propriétés faciles à démontrer.

Proposition 2. *Soit e un endomorphisme d'un groupe $\mathcal{G} = (E, \bigcirc)$ d'élément neutre 0. Pour tout $a \in E$, on note le symétrique de a par $\bar{a}$. L'endomorphisme e a les propriétés suivantes :*

1. $e(0) = 0$;
2. *pour tout* $a \in E$, $e(\bar{a}) = \overline{e(a)}$.

Un *groupe abélien* $\mathcal{A} = (E, \oplus)$ (aussi appelé *groupe commutatif*) est un groupe dont la loi de composition $\oplus$ est commutative sur E. En d'autres termes, pour tout $a, b \in E$, $a \oplus b = b \oplus a$. Dans la suite du chapitre, on notera 0 l'élément neutre d'un groupe abélien et $-a$ le symétrique de $a \in E$. De plus, on notera simplement la loi de composition d'un groupe abélien $+$ (et on utilisera $\sum$ comme pour l'addition normale).

Un *anneau* (*ring* en anglais) $\mathcal{R}$ est composé d'un ensemble E, d'une loi d'addition $\oplus$ telle que le groupe $(E, \oplus)$ est abélien et une loi de multiplication $\otimes$ qui respectent les propriétés suivantes :

— elle est *associative* ;

— elle est *distributive* par rapport à la loi d'addition : pour tout $a, b, c \in E$, $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$;

— elle possède un élément neutre (que l'on notera 1 dans la suite).

Similairement à l'addition, on simplifiera la notation $a \otimes b$ en ab.

Un *corps fini* (*finite field* en anglais) est un anneau $(\mathbb{A}, +, *)$ tel que le groupe $(\mathbb{A} \setminus \{0\}, *)$ (avec 0 l'élément neutre de l'addition) est un groupe abélien. La taille q de l'alphabet d'un corps fini est toujours une puissance d'un nombre premier et un corps fini est toujours commutatif. On note souvent $\mathrm{GF}(q)$ un corps fini de cardinalité q. Quand q est premier, le corps fini $\mathrm{GF}(q) = (\mathbb{A}, +, *)$ se comporte exactement comme $\mathbb{Z}/q\mathbb{Z}$. En d'autres termes, à un renommage des lettres près, les lois d'addition et de multiplication sont des simples additions et multiplications modulo q.

Considérons un groupe $\mathcal{G} = (\mathbb{A}, +)$ et un anneau $\mathcal{R} = (\mathbb{A}, +, *)$. Pour tout $n \in \mathbb{N}$, on définit le groupe $\mathcal{G}^n = (\mathbb{A}^n, \oplus)$ et l'anneau $\mathcal{R}^n = (\mathbb{A}^n, \oplus, \otimes)$ comme suit. La loi d'addition $\oplus$ (*resp.* de multiplication $\otimes$) correspond à la loi $+$ (*resp.* $*$) appliquée composante par composante. Par exemple, pour tout $x, y \in \mathbb{A}^n$, $x \oplus y = (x_1 + y_1, \ldots, x_n + y_n)$.

Dans la suite, plutôt que de dire que l'on considère un groupe, un anneau, *etc.* portant sur l'ensemble $\mathbb{A}$, on dira souvent que $\mathbb{A}$ est *doté* des lois qui correspondent à la structure correspondante.

4.2.2 Matrice

Dans cette sous-section, on considère que l'alphabet $\mathbb{A}$ est doté d'une structure de corps fini.

Une *matrice* $M \in \mathbb{A}^{n \times m}$ peut se définir comme une suite de nm lettres $M(i, j) \in \mathbb{A}$ réparties sur n lignes et m colonnes. Pour tout $i \in [n]$ et $j \in [m]$, $M(i, j)$ désigne la lettre située à la i-ème ligne et à la j-ème colonne. Par exemple, si $M = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$, alors on a $M(1, 1) = 0$, $M(1, 2) = 1$, $M(2, 1) = 2$ et $M(2, 2) = 3$. Si $n = m$ on dit que M est une *matrice carrée*.

Considérons deux ensembles $I = \{i_1, \ldots, i_\ell\} \subseteq [n]$ et $J = \{j_1, \ldots, j_p\} \subseteq [m]$ avec $i_1 < \cdots < i_\ell$ et $j_1 < \cdots < j_p$. La sous-matrice $M(I, J)$ est la matrice $N \in \mathbb{A}^{\ell \times p}$ telle que, pour tout $a \in [\ell]$ et $b \in [p]$, on a $N(a, b) = M(i_a, j_b)$. Si $I = \{i\}$, on peut écrire $M(i, J)$ à la place de $M(I, J)$. De la même façon, si $J = \{j\}$, alors $M(I, j) = M(I, J)$. En particulier, $M\Big([n], j\Big) \in \mathbb{A}^{n \times 1}$ est la matrice composée de la j-ème colonne de M. De même, $M(i, [m]) \in \mathbb{A}^{1 \times m}$ est la matrice composée de la i-ème ligne de M. Considérons deux matrices $M \in \mathbb{A}^{n \times p}$ et $M' \in \mathbb{A}^{p \times m}$. On note $N = MM'$ le *produit* de ces deux matrices. Il s'agit de la matrice $N \in \mathbb{A}^{nm}$ telle que, pour tout $i \in [n]$ et pour tout $j \in [m]$, on a :

$$N(i, j) = \sum_{k=1}^{p} M(i, k) M'(k, j).$$

Remarquons que même pour des matrices carrées, le produit matriciel n'est pas nécessairement commutatif. En d'autres termes, on peut avoir $MM' \neq M'M$. En revanche, le produit des matrices et associatif : $(MM')M'' = M(M'M'')$.

Considérons maintenant deux matrices de même dimensions $M, M' \in \mathbb{A}^{n \times m}$. Le *produit (matriciel) de Hadamard* de ces deux matrices, noté $M \odot M'$ est la matrice $N \in \mathbb{A}^{n \times m}$ telle que, pour tout $i \in [n]$ et pour tout $j \in [m]$, on a $N(i, j) = M(i, j)M'(i, j)$.

On appelle *diagonale principale* d'une matrice carrée $M \in \mathbb{A}^{n \times n}$ le tuple composé des entrées $M(i, i)$ avec $i \in [n]$. La matrice identité I_n est une matrice carrée de taille $n \times n$ qui est composée de `1` sur toute sa diagonale principale et de `0` partout ailleurs. On écrit I en lieu et place de I_n quand la valeur n est claire dans le contexte. Une propriété bien connue de la matrice identité est que, pour toute matrice $M \in \mathbb{A}^{m,n}$, on a $I_m M = M I_n = M$.

Si toutes les entrées en dessous de la diagonale d'une matrice carrée $M \in \mathbb{A}^{n \times n}$ sont nulles (*i.e.* pour tout $i < j \in [n]$, $M(i, j) = 0$), on dit que M est une *matrice triangulaire supérieure*. Si toutes celles en dessous de la diagonale le sont, on dit que M est une *matrice triangulaire inférieure*.

Une matrice *inversible* (ou *non singulière*), est une matrice carrée $M \in \mathbb{A}^{n \times n}$ telle qu'il existe une matrice $M^{-1} \in \mathbb{A}^{n \times n}$ (appelé *inverse* de M) telle que $MM^{-1} = I_n$. Notons que, pour tout $M, M' \in \mathbb{A}^{n \times n}$, on a la double équivalence

$$MM' = I_n \iff M'M = I_n \iff M' = M^{-1}.$$

Le déterminant d'une matrice carrée $M \in \mathbb{A}^{n \times n}$ peut être défini à partir de la formule de Leibniz :

$$\det(M) := \sum_{\pi \in \mathbb{S}_{[n]}} \varepsilon(\pi) \prod_{i \in [n]} M(i, \pi(i)),$$

avec

$$\varepsilon(\pi) := \prod_{1 \leq i < j \leq n} \operatorname{sgn}(\pi(i) - \pi(j)) \text{ et } \operatorname{sgn} : t \mapsto \begin{cases} 1 & \text{si } t \geq 0 \\ -1 & \text{sinon} \end{cases}.$$

En particulier si $M \in \mathbb{A}^{1\times 1}$ et que $M(1,1) = a$ alors $\det(M) = a$.

On peut aussi calculer récursivement le déterminant de M en *développant selon une ligne* $i \in [n]$. On a alors,

$$\det(M) = \sum_{j\in[n]} M(i,j)(-1)^{i+j} M\Big([n] \setminus \{i\}, [n] \setminus \{j\}\Big).$$

On peut aussi *développer un déterminant selon une colonne* $j \in [n]$ de manière totalement similaire. Une *matrice de permutation* est une matrice carrée M qui a une entrée `1` sur chaque ligne et sur chaque colonne et des `0` partout ailleurs. On peut vérifier que le déterminant d'une telle matrice est 1 ou -1. En particulier, $\det(I_n) = 1$ pour tout n.

On rappelle que l'on suppose que l'alphabet $\mathbb{A}$ est doté d'une structure de corps fini (ce qui suit n'est pas vrai pour tout anneau). Une matrice est inversible si et seulement si son déterminant est non nul. En outre, pour tout $M, M' \in \mathbb{A}^n$, on a l'égalité $\det(MM') = \det(M)\det(M')$. Il en résulte que si M et M' sont inversibles alors MM aussi. Le rang d'une matrice $M \in \mathbb{A}^{n\times m}$ (noté $\mathrm{rg}(M)$) est le minimum entre le nombre de colonnes et le nombre de lignes linéairement indépendantes. Une matrice de *rang plein* est une matrice carrée $M \in \mathbb{A}^{n\times n}$ de rang n.

En outre, quand on ne travaille pas sur un corps fini mais sur un anneau quelconque, une matrice est inversible si est seulement si $\det(M)$ est inversible.

4.2.3 Matrice d'adjacences et familles de réseaux d'automates

Considérons un graphe d'interaction $G = (V, A)$ de taille n. On appelle la matrice d'adjacence de G la matrice $M_G \in \{\mathtt{0}, \mathtt{1}\}^{n\times n}$ telle que

$$M_G(i,j) = \begin{cases} 1 & \text{si } (i,j) \in A, \\ 0 & \text{sinon.} \end{cases}$$

On note $\mathbb{M}(G, q)$ l'ensemble des matrices $M \in \mathbb{A}^{n\times n}$ telles que $M(i,j) \neq 0$ si et seulement si $(i,j) \in A$.

Dans ce chapitre, on va s'intéresser à des familles de réseaux d'automates particulières. On les donne par ordre décroissant de généralités :

1. un réseau d'automates $f \in \mathrm{F}(n,q)$ est dit *abélien* si f est un endomorphisme du groupe $\mathcal{A}^n$ avec $\mathcal{A}$ un groupe abélien sur $\mathbb{A}$. Plus concrètement, pour tout $i \in [n]$, on a $f_i : x \mapsto \sum_{j=1}^n e_{i,j}(x_j)$ avec $e_{i,j}$ un endomorphisme de $\mathcal{A}$ tel que $e_{i,j}(x_j) = f_i(x_j e^j)$;
2. un réseau d'automates $f \in \mathrm{F}(n,q)$ est *linéaire* si on a $f(x) = xM$, avec $M \in A^{n\times n}$ selon la loi d'addition et de multiplication d'un anneau $\mathcal{R}$ sur $\mathbb{A}$;
3. en particulier, si $\mathcal{R}$ est un corps fini alors $f \in \mathrm{F}(n,q)$ est *corps fini linéaire* ;

4. un réseau d'automates $f \in \mathrm{F}(n,2)$ est le *réseau XOR* de G si $f : x \mapsto xM_G$. Tout réseau abélien de $\mathrm{F}(n,2)$ peut être vu comme un réseau XOR pour un certain digraphe G.

4.2.4 Trace et réseaux d'automates expansifs

Fixons $f \in \mathrm{F}(n,q)$. Pour tout $x \in \mathbb{A}^n$ et $i \in [n]$, la *trace* de x sur i est la séquence infinie

$$\rho_i(x) = (f_i(x), f_i^2(x), \dots).$$

Comme dans l'introduction, on note $\rho_i^{(t)}(x) = (f_i(x), \dots, f_i^t(x))$ les t premières valeurs de cette trace. Comme dit plus haut, f est *expansif* si pour toute paire de configurations $\{x,y\} \subseteq \mathbb{A}^n$ et tout $i \in [n]$, il existe un entier $t \geq 1$ tel que $f_i^t(x) \neq f_i^t(y)$. De manière équivalente, f est expansif si et seulement si pour tout $i \in [n]$, il existe un entier $s \geq 1$ tel que la fonction $\rho_i^{(s)}$ est injective. Si f est abélien, alors la fonction locale f_i aussi et par conséquent la fonction $\rho_i^{(s)} : \mathbb{A}^n \to \mathbb{A}^s$ également. De ce fait, si f est abélien alors f est expansif si et seulement pour tout $x \in \mathbb{A}^n \setminus \{0^n\}$ et pour tout $i \in [n]$, il existe un entier $t \geq 1$ tel que $f_i^t(x) \neq f_i^t(0^n)$. En effet, pour tout $x, y \in \mathbb{A}^n$, $f_i^t(x) \neq f_i^t(y)$ si et seulement si $f_i^t(x-y) \neq f_i^t(0^n)$. Soit f un réseau linéaire (*i.e.* $f(x) = xM$). Considérons les n puissances de M : $M^0 = I, M^1 = M, \dots, M^n$. Pour tout $t \geq 0$ et tout $j \in [n]$, construisons la matrice $N_j^{(t)} \in \mathbb{A}^{n \times n}$ suivante :

$$N_j^{(t)} := N_j^{[t,t+n[} := \Big(\; M^t\Big([n],j\Big) \;\Big|\; M^{t+1}\Big([n],j\Big) \;\Big|\; \dots \;\Big|\; M^{t+n-1}\Big([n],j\Big) \; \Big).$$

La matrice $N_j^{(t)}$ nous permet de déterminer si f est expansif quand f est un corps fini linéaire.

Lemme 15. *Les propriétés suivantes sont équivalentes pour un réseau corps fini linéaire $f : x \mapsto xM$:*

1. *f est expansif ;*
2. *M est inversible et pour tout $j \in [n]$, $N_j^{(0)}$ est inversible ;*
3. *pour tout $t \geq 1$ et $j \in [n]$, $N_j^{(t)}$ est inversible ;*
4. *il existe $t \geq 1$ tel que, pour tout $j \in [n]$, $N_j^{(t)}$ est inversible.*

Démonstration. Prouvons que la propriété 2 implique la propriété 3. On peut voir que, pour tout t et k, $M^{t+k}\Big([n],j\Big) = M^t\Big(M^k([n],j)\Big)$. En effet, « récupérer » la colonne j d'une matrice $\mathbb{A}^{n\times n}$ revient à multiplier par la matrice $C_j \in \mathbb{A}^{n\times 1}$ définie comme suit. On a $C_j(j,1) = 1$ et pour tout $i \neq j$, on a $C_j(i,1) = 0$. La multiplication de matrice étant associative, on a

$$M^{t+k}\Big([n],j\Big) = (M^tM^k)C_j = M^t(M^kC_j) = M^t\Big(M^k\Big([n],j\Big)\Big).$$

On peut montrer que $N_j^{(t)} = M^t N_j^{(0)}$. En effet, pour tout $j' \in [n]$,

$$N_j^{(t)}\Big([n], j'\Big) = M^{t+j'-1}\Big([n], j'\Big) = M^t\Big(M^{j'-1}\Big([n], j'\Big)\Big) = M^t(N_j^{(0)}([n], j')).$$

Par conséquent, si M et $N_j^{(0)}$ sont inversibles, alors $N_j^{(t)}$ aussi.

Clairement, la propriété 3 implique la propriété 4.

Prouvons que la propriété 4 implique la propriété 1. Supposons que la propriété 4 soit vraie. Définissons alors le mot $y = (f_j^t(x), \ldots, f_j^{t+n-1}(x)) = xN_j^{(t)}$. Donc, $x = y(N_j^{(t)})^{-1}$ peut être retrouvé depuis le mot y et le réseau f est expansif.

Prouvons maintenant que la propriété 1 implique la propriété 2. Clairement, si le réseau f est expansif, alors il est également bijectif. La matrice M est donc inversible. Supposons que le réseau f soit expansif, mais que la matrice $N_j^{(0)}$ soit non inversible pour un certain entier $j \in [n]$. Par expansivité, il existe un entier $s > n$ tel que la matrice

$$N_j^{[0,s[} = \Big(\; M^0\Big([n], j\Big) \;\Big|\; M^1\Big([n], j\Big) \;\Big|\; \ldots \;\Big|\; M^{s-1}\Big([n], j\Big) \; \Big),$$

soit de rang plein. On a supposé que la matrice $N_j^{(0)} = N_j^{[0,n[}$ était non inversible. Il existe donc un entier $i < n$ tel que la j-ème colonne de M^i $(= N_j^{[0,i[}([n], i))$ est une combinaison linéaire des colonnes de $N_j^{[0\ldots i[}$. Posons $M^i\Big([n], j\Big) = \sum_{\ell=0}^{i-1} y_\ell M^\ell([n], j)$. Donc, pour tout $k \geq 0$,

$$M^{\ell+k}([n], j) = M^k M^\ell([n], j) = M^k \sum_{\ell=0}^{i-1} y_\ell M^\ell([n], j) = \sum_{\ell=0}^{i-1} y_\ell M^{\ell+k}([n], j).$$

On a donc l'inégalité

$$\mathrm{rg}(N_j^{[0,s[}) = \mathrm{rg}(N_j^{[0,i[}) \leq \mathrm{rg}(N_j^{(0)}) < n.$$

C'est une contradiction. Par conséquent, $\mathrm{rg}(N_j^{(0)}) = n$ et la matrice $N_j^{(0)}$ est inversible. □

Calculer le déterminant d'une matrice n'est pas plus difficile que de multiplier des matrices (voir le théorème 6.6 de [1]). Le lemme 15 ci-dessus nous donne donc un algorithme efficace pour déterminer l'expansivité d'un réseau d'automates corps fini linéaire.

Corolaire 3. *On peut déterminer si un réseau d'automates corps fini linéaire est expansif en $\mathcal{O}(n \cdot M(n))$, avec $M(n)$ le temps nécessaire à la multiplication de deux matrices carrées $n \times n$.*

4.3 Graphes d'interaction et expansivité

Cette section est consacrée à la caractérisation des graphes d'interaction qui admettent un réseau expansif. En d'autres termes, on s'intéresse aux graphes d'interaction G pour lesquels il existe une taille d'alphabet q telle qu'il existe un réseau $f \in \mathrm{F}(G, q)$ expansif.

On peut facilement se convaincre qu'un réseau expansif est également bijectif. En effet, si f n'est pas bijectif, alors il existe deux configurations distinctes $x, y \in \mathbb{A}^n$ telles que $f(x) = f(y)$. Clairement, pour tout $i \in [n]$, et pour tout $t \geq 1$, on aura $f_i^t(x) = f_i^t(y)$ et donc f n'est pas expansif. On étudie d'abord des résultats sur l'existence de réseaux d'automates bijectifs. On appelle une *décomposition en cycles* d'un digraphe $G = (V, A)$ un ensemble de cycles disjoints dans G qui partitionnent l'ensemble V. On dit qu'un digraphe est *couvrable* s'il a une décomposition en cycle. La proposition 3.11.6 de [17], nous donne le lemme suivant (qui découle essentiellement du théorème du mariage de Hall).

Lemme 16 ([17]). *Le digraphe $G = (V, A)$ est couvrable si et seulement si $|\mathcal{N}^{out}(S)| \geq |S|$ pour tout $S \subseteq V$.*

De plus, le corolaire 4 de [65] montre que l'on peut aussi caractériser ces graphes via l'existence de réseaux d'automates bijectifs.

Théorème 11 ([65]). *Le digraphe G est couvrable si et seulement si, pour tout $q \geq 3$, $\mathrm{F}(G, q)$ contient une bijection.*

Le théorème 12 montre un résultat analogue en remplaçant les fonctions bijectives par des fonctions corps fini linéaires.

Théorème 12. *Si G est couvrable, alors $\mathbb{M}(G, q)$ contient une matrice inversible pour toute puissance d'un nombre premier $q \geq 3$.*

Démonstration. Soit $q \geq 3$. Supposons d'abord que $G = (V, A)$ soit boucle-complet, c'est-à-dire qu'il ait une boucle sur chaque sommet. Pour tout $k \in [n]$, soit G_k le sous-graphe de G restreint aux k premiers sommets. Plus formellement, $G_k = \big([k], A_k\big)$ avec $A_k = \{(i, j) \in A \mid i, j \in [k]\}$. Prouvons par récurrence que, pour tout $k \in [n]$, il existe une matrice $M \in \mathbb{M}(G_k, q)$ dont le déterminant est $\mathtt{1}$. Clairement si $n = 1$, on peut prendre $M \in \mathbb{A}^{1\times 1}$ avec $M(1, 1) = \mathtt{1}$ et on a bien $\det(M) = 1$.

Maintenant, supposons que le résultat est vrai pour $k-1 \in [n-1]$. Soit $M' \in \mathbb{M}(G_{k-1}, q)$ de déterminant 1. Alors, définissons la matrice $N \in \mathbb{A}^{k\times k}$ comme suit :

$$N(i, j) = \begin{cases} M'(i, j) & \text{si } i, j \in [k-1], \\ \mathtt{1} & \text{si } (i, j) \in A \text{ et } (i = k \text{ ou } j = k), \\ \mathtt{0} & \text{sinon.} \end{cases}$$

En particulier, comme il y a une boucle $(k,k) \in A$, on a $N(k,k) = 1$. On remarque que $N \in \mathbb{M}(G_k, q)$.

Si $\det(N) \neq 2$, alors considérons la matrice $M \in \mathbb{M}(G_k, q)$ telle que

$$M(i,j) = \begin{cases} 2 - \det(N) & \text{si } i = j = k, \\ N(i,j) & \text{sinon.} \end{cases}$$

Calculons les déterminants de N et M en les développant suivant la ligne k.

$$\begin{aligned}
\det(N) &= \sum_{j=1}^{k} N(k,j)(-1)^{k+j} \det\Big(N\Big([k-1], [k] \setminus \{j\}\Big)\Big) \\
&= N(k,k)(-1)^{2k} \det(M') + \sum_{j=1}^{k-1} N(k,j)(-1)^{k+j} \det\Big(N\Big([k-1], [k] \setminus \{j\}\Big)\Big) \\
&= \det(M') + s = 1 + s, \\
\text{avec } s &= \sum_{j=1}^{k-1} N(k,j)(-1)^{k+j} \det\Big(N\Big([k-1], [k] \setminus \{j\}\Big)\Big), \\
\det(M) &= \sum_{j=1}^{k} M(k,j)(-1)^{k+j} \det\Big(M\Big([k-1], [k] \setminus \{j\}\Big)\Big) \\
&= M(k,k)(-1)^{2k} \det(M') + \sum_{j=1}^{k-1} M(k,j)(-1)^{k+j} \det\Big(M\Big([k-1], [k] \setminus \{j\}\Big)\Big) \\
&= M(k,k) \det(M') + \sum_{j=1}^{k-1} N(k,j)(-1)^{k+j} \det\Big(N\Big([k-1], [k] \setminus \{j\}\Big)\Big) \\
&= M(k,k) + s.
\end{aligned}$$

On a donc $\det(M) = \det(N) - 1 + M(k,k) = \det(N) - 1 + (2 - \det(N)) = 1$. Si $\det(N) = 2$, alors considérons la matrice $M \in \mathbb{M}(G_k, q)$ telle que

$$M(i,j) = \begin{cases} 2 & \text{si } i = j = k, \\ -N(i,j) & \text{si } i = k \text{ et que } j \leq k, \\ N(i,j) & \text{sinon.} \end{cases}$$

Calculons le déterminant de la matrice M en le développant suivant la ligne k.

$$\begin{aligned}
\det(M) &= \sum_{j=1}^{k} M(k,j)(-1)^{k+j} \det\Big(M\Big([k-1],[k]\setminus\{j\}\Big)\Big) \\
&= M(k,k)(-1)^{2k}\det(M') + \sum_{j=1}^{k-1} M(k,j)(-1)^{k+j}\det\Big(M\Big([k-1],[k]\setminus\{j\}\Big)\Big) \\
&= M(k,k)\det(M') - \sum_{j=1}^{k-1} N(k,j)(-1)^{k+j}\det\Big(N\Big([k-1],[k]\setminus\{j\}\Big)\Big) \\
&= M(k,k) - s.
\end{aligned}$$

On a $-det(N) = -1 - s$ et donc $\det(M) = -\det(N) + 1 + M(k,k) = -2 + 1 + 2 = 1$.

Par récurrence, pour tout $k \in [n]$, il existe une matrice $M \in \mathbb{M}(G_k, q)$ de déterminant 1. En particulier, comme $G_n = G$, il existe une matrice $M \in \mathbb{M}(G, q)$ telle que $\det(M) = 1$. Ceci conclut la démonstration pour les digraphes boucle-complets.

Maintenant, pour le cas plus général, supposons que G est couvrable. Il existe donc une décomposition en cycles de G. Soit $\pi : V \to V$ la permutation telle que, pour tout $i \in V$, $\pi(i)$ est le successeur de i dans le cycle auquel appartient i. Clairement π est bijective. Notons P la matrice de permutation associée à π. On a donc, pour tout $i, j \in [n]$, $P(i,j) = \begin{cases} \mathtt{1} & \text{si } i = \pi(j) \\ \mathtt{0} & \text{sinon} \end{cases}$. On remarque que, dans une colonne j de P, seule la case $(\pi(j), j)$ est non nulle. On peut faire la même remarque pour les lignes. On peut définir l'inverse P^{-1} de la matrice P de la façon suivante : pour tout $i, j \in [n]$, $P^{-1}(i,j) = \begin{cases} \mathtt{1} & \text{si } i = \pi^{-1}(j) \\ \mathtt{0} & \text{sinon} \end{cases}$.

Soit $G' = (V', A')$ le digraphe défini par sa matrice d'adjacence $M_{G'} = P^{-1}M_G$. On peut se convaincre que G' est boucle-complet. En effet, pour tout $i \in [n]$,

$$M_{G'}(i,i) = \sum_{k=1}^{n} P^{-1}(i,k)M_G(k,i) = P^{-1}(i,\pi^{-1}(i))M_G(\pi^{-1}(i),i) = \mathtt{1}.$$

Par conséquent, G' est boucle-complet, et donc, comme prouvé précédemment, il existe une matrice $M' \in \mathbb{M}(G', q)$ telle que $\det(M') = 1$. La matrice P est une matrice de permutation et son déterminant est donc $\mathtt{1}$ ou $-\mathtt{1}$.

On définit la matrice $M = PM'$ si $\det(P) = 1$ et $M = -PM'$ sinon. Dans les deux cas, on a bien $\det(M) = 1$. Il suffit maintenant de prouver que $M \in \mathbb{M}(G, q)$. Montrons d'abord que $(i,j) \in A$ si et seulement si $(\pi^{-1}(i), j) \in A'$. Comme, $M_{G'} = P^{-1}M_G$, on a

$$M_{G'}(\pi^{-1}(i),j) = \sum_{k=1}^{n} P^{-1}(\pi^{-1}(i),k)M_G(k,j) = P^{-1}(\pi^{-1}(i),i)M_G(i,j) = M_G(i,j).$$

On a donc $M_G(i,j) \neq 0 \iff M_{G'}(\pi^{-1}(i),j) \neq 0$ et donc $(i,j) \in A \iff (\pi^{-1}(i),j) \in A'$. Maintenant, sans perte de généralité, on considère que $M = PM'$. Pour tout $i,j \in [n]$,

$$M(i,j) = \sum_{k=1}^{n} P(i,k)M'(k,j) = P(i,\pi^{-1}(i))M'(\pi^{-1}(i),j) = M'(\pi^{-1}(i),j).$$

On sait que $M' \in \mathbb{M}(G',q)$. Donc,

$$M(i,j) \neq 0 \iff M'(\pi^{-1}(i),j) \neq 0 \iff (\pi^{-1}(i),j) \in A' \iff (i,j) \in A.$$

Par conséquent, $M \in \mathbb{M}(G,q)$. Or, $\det(M) = 1$ et la matrice M est donc inversible ce qui conclut la démonstration.

□

Certains digraphes admettent des réseaux expansifs pour tout alphabet (voir la section 4.4). D'autres n'admettent aucun réseau expansif, quel que soit l'alphabet. Nous allons maintenant caractériser les digraphes qui admettent des réseaux expansifs pour un certain alphabet. En fait, on peut être plus précis et considérer plusieurs variations de la définition d'un réseau expansif sans changer la caractérisation des graphes d'interaction qui les admettent.

Définition 4. *Un réseau d'automates $f \in \mathrm{F}(n,q)$ est* faiblement expansif *si, pour toute paire de configurations $\{x,y\} \subseteq \mathbb{A}^n$ et $i \in [n]$, il existe un entier $t \geq 0$ tel que $f_i^t(x) \neq f_i^t(y)$.*

Notons qu'un réseau f faiblement expansif n'est pas nécessairement bijectif.

Définition 5. *Un réseau d'automates f est* quasi-expansif *si, pour toute paire de configurations $\{x,y\} \subseteq \mathbb{A}^n$ et $i \in [n]$, il existe un entier $t \geq 0$ tel que $f_{\mathrm{N}(i)}^t(x) \neq f_{\mathrm{N}(i)}^t(y)$.*

Cette dernière définition est celle qui correspond aux automates cellulaires (voir la section 4.9). Il n'est pas difficile de voir que ces définitions ne sont pas équivalentes. En revanche, les graphes d'interaction qu'elles caractérisent sont les mêmes comme le montre le théorème suivant.

Théorème 13. *Les propriétés suivantes sont équivalentes pour un graphe d'interaction G de taille $n \geq 2$:*

1. *G est fortement connexe et couvrable ;*
2. *$\mathrm{F}(G,q)$ contient un réseau f expansif pour un certain q ;*
3. *$\mathrm{F}(G,q)$ contient un réseau f quasi-expansif pour un certain q ;*
4. *$\mathrm{F}(G,q)$ contient un réseau f faiblement expansif pour un certain q ;*
5. *G admet un réseau f linéaire expansif sur un corps fini suffisamment grand.*

Clairement, un réseau expansif est quasi-expansif et faiblement expansif. Par conséquent, la propriété 2 implique les propriétés 3 et 4. Encore plus clairement, la propriété 5 implique la propriété 2. Pour prouver que ces 5 propriétés sont bien équivalentes on va prouver dans le lemme 17 que la propriété 3 implique la propriété 1, dans le lemme 18 que la propriété 4 implique la propriété 1, et enfin dans le théorème 18, que la propriété 1 implique la propriété 5.

Lemme 17. *Si $G = (V, A)$ admet un réseau quasi-expansif, alors G est fortement connexe et couvrable (ou $|V| = 1$).*

Démonstration. Soit $f \in \mathrm{F}(G, q)$ pour un certain $q \geq 2$. Supposons d'abord que $G = (V, A)$ n'est pas fortement connexe. Considérons les sommets i et $j \in V$ tels qu'il n'y a pas de chemin entre i et j dans G. Clairement, il n'y a pas de chemin entre i et $\mathcal{N}(j)$ non plus. On peut alors prendre $x, y \in \mathbb{A}^n$ qui différent seulement sur i (par exemple, on peut prendre $x_{V\setminus\{i\}} = (\mathtt{0})^{n-1} = y_{V\setminus\{i\}}$ et $x_i = \mathtt{0} \neq \mathtt{1} = y_i$). On a bien $x \neq y$ si $n \geq 2$. Clairement, pour tout $t \geq 0$, on a $f^t_{\mathcal{N}(j)}(x) \neq f^t_{\mathcal{N}(j)}(y)$. Par conséquent, f n'est pas quasi-expansif ce qui est une contradiction. Donc, si f n'est pas quasi-expansif alors G est fortement connexe.

Supposons maintenant que G n'est pas couvrable. D'après le lemme 16, il existe un ensemble $S \subseteq V$ tel que $|\mathcal{N}^{\mathrm{out}}(S)| < |S|$. Soit $j \in V \setminus \mathcal{N}^{\mathrm{out}}(S)$ (j peut être dans S ou non). La fonction $\widetilde{f}_{\mathcal{N}^{\mathrm{out}}(S)} : A^{|S|} \to A^{|\mathcal{N}^{\mathrm{out}}(S)|}$ n'est clairement pas injective (car $|\mathcal{N}^{\mathrm{out}}(S)| < |S|$). Donc, on peut trouver deux configurations $x, y \in \mathbb{A}^n$ distinctes seulement sur S telles que $f_{\mathcal{N}^{\mathrm{out}}(S)}(x) = f_{\mathcal{N}^{\mathrm{out}}(S)}(y)$. Toutefois, il est également vrai que $f_i(x) = f_i(y)$ pour tout $i \notin \mathcal{N}^{\mathrm{out}}(S)$. On a donc $f(x) = f(y)$. Plus généralement, pour tout $t \geq 1$, on a $f^t(x) = f^t(y)$ et $x_{\mathcal{N}(j)} = y_{\mathcal{N}(j)}$ ce qui prouve que f n'est pas quasi-expansif. □

Lemme 18. *Si $G = (V, A)$ admet un réseau faiblement expansif, alors G est fortement connexe et couvrable (ou $|V| = 1$).*

Démonstration. Cette preuve est similaire à celle du lemme 17. De nouveau, il est clair que G doit être fortement connexe. Si G n'est pas couvrable, alors considérons un ensemble $S \subseteq V$ tel que $|\mathcal{N}^{\mathrm{out}}(S)| < |S|$. Si $S = V$, alors tout sommet $i \in V \setminus \mathcal{N}^{\mathrm{out}}(V)$ est une source (*i.e.* $\mathcal{N}(i) = \emptyset$). Alors, la fonction locale f_i est une fonction constante $f_i : x \mapsto c$ avec $c \in \mathbb{A}$. Prenons alors les deux configurations $x, y \in \mathbb{A}^n$ telles que

$$x_i = c = y_i \text{ et } x_{V\setminus\{i\}} = (\mathtt{0})^{n-1} \neq (\mathtt{1})^{n-1} = y_{V\setminus\{i\}}.$$

On a alors $f^t_i(x) = f^t_i(y) = c$ pour tout $t \geq 0$ alors que $x \neq y$.

Si $S \neq V$, alors comme pour la preuve précédente, il existe deux configurations $x, y \in \mathbb{A}^n$ distinctes sur S uniquement et telles que $f(x) = f(y)$. Donc, pour tout $j \notin S$ et $t \geq 0$, $f^t_j(x) = f^t_j(y)$. □

Théorème 14. *Tout digraphe $G = (V, A)$ fortement connexe et couvrable de taille n admet un réseau expansif linéaire sur un corps fini $\mathrm{GF}(q)$ avec q une puissance de nombre premier telle que $q \geq \frac{1}{2}(n^3 + n^2 + 4)$.*

Démonstration. On rappelle que d'après les propriétés 3 et 4 du lemme 15, un réseau linéaire $f(x) = xM$ est expansif si et seulement si pour tout $i \in [n]$, la matrice

$$N_i^{(1)} = N_i^{[n]} = \Big(\; M\big([n], i\big) \;\Big|\; M^2\big([n], i\big) \;\Big|\; \dots \;\Big|\; M^n\big([n], i\big) \; \Big),$$

est inversible.

La présente démonstration est non constructive. On peut considérer les coefficients non nuls de la matrice M comme des variables. On va montrer que le déterminant de N est un polynôme non nul de ces variables. On montrera également que si l'alphabet du corps fini considéré est suffisamment grand, alors on peut toujours trouver une valuation des variables qui rend la matrice N inversible.

Soit $\{\bar{C}_1, \dots, \bar{C}_s\}$ un ensemble de cycles qui forme une décomposition en cycles de G. On va associer chaque arc a appartenant à un cycle $\bar{C}_k$ à une variable $x(a) = \bar{\alpha}_k$. On associe tous les autres arcs $a \in A$ du graphe à leurs propres variables $x(a) = \bar{\beta}_a$. En particulier, cela signifie que les cordes des cycles $\bar{C}_1, \dots, \bar{C}_s$ ont leurs propres variables.

Pour tout chemin $w = (a_1, \dots, a_\ell) \in A^\ell$ dans G, on note $X(w)$ le monôme $X(w) = x(a_1)x(a_2)\cdots x(a_\ell)$. Notons que la somme et le produit des variables commutent, car on va les évaluer dans un corps fini.

Fixons i comme un sommet de $\bar{C}_\sigma$. Soit T, un « arbre de cycles » dont la racine est $\bar{C}_\sigma$. Plus formellement, T est un sous-graphe de G contenant tout V. Il contient tous les arcs des cycles $\bar{C}_1, \dots, \bar{C}_s$ et pour tout $k \neq \sigma$, il y a exactement un arc sortant dans T qui quitte $\bar{C}_k$. Le cycle $\bar{C}_\sigma$, quant à lui, n'a pas d'arc sortant dans T. Notons qu'un tel « arbre de cycle » peut facilement être construit. En effet, on peut contracter chaque cycle en un sommet. Ensuite, il ne reste qu'à construire un arbre couvrant dont la racine est le sommet correspondant au cycle $\bar{C}_\sigma$, en utilisant la forte connexité de G.

Réordonnons maintenant les cycles $\bar{C}_1, \dots, \bar{C}_s$ selon l'ordre topologique de T. Nous avons donc $C_1 = \bar{C}_\sigma, C_2, \dots, C_s$. De la même façon, pour tout $k \in [s]$, on définit la variable $\alpha_k := \bar{\alpha}_{k'}$ comme étant la variable associée à tous les arcs de $C_k = \bar{C}_{k'}$. De la même façon, pour tout $k \in [2, s]$, la variable β_k correspond à la variable de l'arc quittant C_k dans T.

Pour tout $k \in [s]$, soit ℓ_k la taille du cycle C_k et $\Lambda_k := \ell_1 + \cdots + \ell_{k-1}$ ($\Lambda_1 = 0$) la somme des tailles des cycles $C_1, \dots, C_k$. On note w_k le plus court chemin entre C_k et i dans T. On note λ_k la taille de w_k et on note $X_k = X(w_k)$.

Pour $k = 1$, w_k est un chemin de taille $\lambda_1 = 0$ et $X_1 = 1$. Il est facile de voir que $\Lambda_k \geq \lambda_k$ pour tout $k \in [s]$.

On rappelle que $d_T(i, j)$ est la distance entre i et j dans T. Nous remarquons que, pour tout $j, j' \in C_k$ distincts, $d_T(j, i) \not\equiv d_T(j', i) \mod \ell_k$.

On peut alors noter les sommets d'un cycle C_k en fonction de leurs distances à i comme

suit. On nomme les sommets de C_k par j_k^p avec $p \in [\ell_k]$ de façon à respecter

$$d_T(j_k^p, i) \equiv \Lambda_k + p \mod \ell_k.$$

Notons $W(j,k,t)$ l'ensemble des chemins dans G entre j et k de taille t.

Affirmation 4. *Prenons $k \in [n]$. Pour toute ligne (qui correspond à un sommet) $j \in V$ et toute colonne (temps) $t \in [n]$, on peut montrer que l'on a*

$$N_k(j,t) = \sum_{w \in W(j,k,t)} X(w).$$

Démonstration. On peut le prouver par récurrence. D'abord, c'est vrai pour $t = 1$. En effet, les chemins d'arcs de taille 1 sont composés d'un unique arc. Le chemin w formé de l'arc $a = (j,k)$ existe si et seulement si $a \in A$. Si $a \notin A$ alors $N_k(j,1) = M(j,k) = 0$. Sinon, $N_k(j,1) = M(j,k) = X(w) = x(a)$.

Supposons maintenant que l'affirmation soit vraie pour $t-1$. On peut décomposer chaque chemin w de j à k de taille t en un arc $(j,j') \in A$ et un chemin w' de j' à k de taille $t-1$. Toute entrée $M^{t-1}(j',k) = N_k(j',t-1)$ a comme valeur le polynôme $\sum_{w' \in W(j',k,t-1)} X(w')$. On a donc l'égalité suivante :

$$N_k(j,t) = M^t(j,k) = \sum_{(j,j') \in A} \prod_{w' \in W(j',k,t-1)} x(a)X(w') = \sum_{w \in W(j,k,t)} X(w).$$

■

Notons $N = N_i^{(1)}$. On a donc pour tout $j \in V$ et $t \in [n]$,

$$N(j,t) = \sum_{w \in W(j,i,t)} X(w).$$

Considérons $j = j_k^p$ et $t = \Lambda_k + p$. Il y a un chemin canonique w dans $W(j,i,t)$: parcourir le cycle C_k autant de fois que possible et ensuite prendre w_k qui est le plus court chemin entre C_k et i dans T. On rappelle que $X_k = X(w_k)$. D'après l'affirmation 4, le chemin produit un terme $X(w) = \alpha_k^{t-\lambda_k} X_k$ dans la case $N_k(j,t)$.

Si on observe les autres chemins de $W(j,i,t)$, on s'aperçoit que soit ils ne restent pas dans T tout le long, soit ils restent dans T, mais ils n'utilisent pas l'arc α_k autant de fois que w. Ceci donne

$$N(j_k^p, \Lambda_k + p) = \alpha_k^{\Lambda_k + p - \lambda_k} X_k + \Gamma + \Delta,$$

avec Γ un polynôme dont chaque terme a une variable β_a d'un arc a en dehors de T et Δ un polynôme dont chaque terme a un degré sur la variable α_k d'au plus $\Lambda_k + p - 1$ (*i.e.*

inférieur à celui de $X(w)$). De ce fait, chaque produit $\prod_{p\in[\ell_k]} N(j_k^p, \Lambda_k+p)$ contient le monôme $\alpha_k^{d_k} X_k^{\ell_k}$ avec $d_k := \sum_{p\in[\ell_k]} \Lambda_k + p - \lambda_k = \ell_k\left(\frac{1}{2}(\ell_k+1) + \Lambda_k - \lambda_k\right)$. Par conséquent, le produit $\prod_{\substack{k\in[s]\\p\in[\ell_k]}} N(j_k^p, \Lambda_k+p)$ contient le monôme $Y = \prod_{k\in[s]} \alpha_k^{d_k} X_k^{\ell_k}$.

On peut calculer le déterminant de N grâce à la formule de Leibniz :

$$\det(N) := \sum_{\pi\in\mathbb{S}_{[n]}} \varepsilon(\pi) \prod_{j\in[n]} N\Big(j, \pi(j)\Big) = \sum_{\pi\in\mathbb{S}_{[n]}} \varepsilon(\pi) \prod_{\substack{k\in[s]\\p\in[\ell_k]}} N\Big(j_k^p, \pi(j_k^p)\Big).$$

Le terme Y contribue au déterminant de N via la permutation $\pi \in \mathbb{S}_{[n]}$, définie comme $\pi : j_k^p \mapsto \Lambda_k + p$. Nous prouvons maintenant que Y n'apparait dans aucun autre produit de variables qui contribue au déterminant de N. Plus, précisément, on va prouver l'affirmation suivante.

Affirmation 5. *Si une permutation $\pi' \in \mathbb{S}_{[n]}$ produit un terme qui ne contient que des variables de T et où pour tout $k \in [s]$, α_k a un degré d_k, alors $\pi = \pi'$.*

Démonstration. Considérons donc une permutation $\pi' \in \mathbb{S}_{[n]}$ qui produit un terme qui ne contient que des variables de T et où pour tout $k \in [s]$, α_k a un degré d_k. Prouvons par récurrence descendante que, pour tout $k \in [s]$, pour tout $k' \geq k$, on a $\pi(j) = \pi'(j)$ pour tout $j \in C_{k'}$.

Prouvons d'abord le cas $k = s$. Posons $\{t_1, \dots, t_{\ell_k}\} = \pi'(C_k) = \{\pi'\Big(j_k^p\Big) \mid p \in [\ell_k]\}$ avec $t_1 < \cdots < t_{\ell_k}$. On veut prouver que, pour tout $p \in [\ell_p]$, $\pi'(j_k^s) = \pi(j_k^s)$. En d'autres termes, on veut prouver que, pour tout $p \in [\ell_p]$, $\pi'(j_k^p) = \Lambda_k + p$. Comme les cycles $C_1, \dots, C_s$ ont été numérotés selon un ordre topologique de T, si $r < r'$, alors il n'y a pas de chemin entre C_r et $C_{r'}$ dans T. En particulier, les chemins de N correspondant aux sommets en dehors de C_s ne contiennent pas α_s. Donc, le degré de α_s maximum n'est atteint que quand on a $\pi'(C_k) = [n - \ell_k, n]$ et est d'au plus $(t_1 - \lambda_s) + \cdots + (t_{\ell_s} - \lambda_s)$. On a donc

$$t_1 + \cdots + t_{\ell_s} - \ell_s\lambda_s \geq d_s = \ell_s\left(\frac{1}{2}(\ell_s+1) + \Lambda_s\right) - \ell_s\lambda_s,$$

ce qui implique que $t_p = \Lambda_s + p$ pour $p \in [\ell_s]$. De plus, le degré de α_s dans $N(j_s^{p'}, \Lambda_s + p)$ est $\Lambda_s + p - \lambda_s$ si et seulement si $p = p'$. Ceci implique que $\pi(j_s^p) = \Lambda_s + p$ pour tout $p \in [\ell_s]$.

L'étape d'itération est similaire. On suppose la propriété vraie pour tout $k' \in [k+1, s]$. On a donc $\bigcup_{k'\in[k+1,s]} \pi(C_{k'}) = [n - \ell_k, n]$. On doit donc choisir $\pi'(j_k^p) = \pi(j_k^p)$ pour tout $p \in [\ell_k]$.

■

On a donc montré que $\det(N_i^{(1)})$ est un polynôme non nul sur les variables $\{x(a) \mid a \in A\}$ car il contient le monôme Y. Son degré est $d := n(n+1)/2$ car tous les monômes sont de ce degré d'après la formule de Leibniz et qu'il contient au moins un monôme. D'après le lemme de Schwartz-Zippel (voir le théorème 7.1.4 du livre [108]), il y a au plus $d(q-1)^{|A|-1}$ valuations des variables $x(a)$ tels que $\det(N_i^{(1)}) = 0$. Ceci signifie qu'il y a au plus $nd(q-1)^{|A|-1}$ valuations des variables $x(a)$ avec $a \in A$ tels que, pour un certain $i \in [n]$, $\det(N_i^{(1)}) = 0$. Comme $q-1 > nd$, nous avons $(q-1)^q > nd(q-1)^{|A|-1}$, et de ce fait, il existe une valuation des variables $x(a)$ avec $a \in A$ telle que, pour tout $i \in [n]$, $\det(N_i^{(1)}) \neq 0$. □

Nous soulignons deux conséquences de notre résultat. En premier lieu, parlons des alphabets pour lesquels un digraphe G fortement connexe et couvrable admet un réseau linéaire expansif. Définissons le *produit cartésien* de deux réseaux $g \in \mathrm{F}(n,q)$ et $h \in \mathrm{F}(n,r)$ comme suit. Considérons l'alphabet $[0,qr[$ comme le produit d'un alphabet dans $\mathbb{A}$ et d'un autre dans $R = [0,r[$. En d'autres termes, $[0,qr[\cong \mathbb{A} \times R = \{(a,b) : a \in \mathbb{A}, b \in R\}$. On peut donc définir $f = g \times h$ comme la fonction $f \in \mathrm{F}(n,qr)$ avec $f(x) = g(y) \times h(z)$, $y \in \mathbb{A}^n$, $z \in R^n$ et $x_i = (y_i, z_i)$ pour tout $i \in [n]$. Certaines propriétés du produit cartésien sont énumérées dans la proposition ci-dessous.

Proposition 3. *Soit $g \in \mathrm{F}(n,q)$, $h \in \mathrm{F}(n,r)$ et $f = g \times h$,*

1. *si g et h sont expansifs, alors f également ;*
2. *si g et h sont linéaires, alors f également ;*
3. *si g et h ont un graphe d'interaction G, alors f également.*

On remarque que la concaténation de deux réseaux corps fini expansifs linéaires n'est un réseau corps fini expansif linéaire que si les taille de leurs deux alphabets sont deux puissances du même nombre premier.

Si G admet un réseau linéaire expansif pour un alphabet de taille q et un autre pour un alphabet de taille r, alors il admet un réseau linéaire expansif pour un alphabet de taille qr.

Corolaire 4. *Pour tout G, l'ensemble des tailles d'alphabets q pour lesquelles il existe un réseau linéaire expansif $f \in \mathrm{F}(G,q)$ a une densité non nulle.*

Démonstration. Dénotons les nombres premiers par $p_1 < p_2 < \ldots$. Disons que p_j est le plus grand nombre premier plus petit que $q := \frac{1}{2}(n^3 + n^2 + 4)$. Pour tout $i \leq j$, soit $d_i = \lceil \log_{p_i}(q) \rceil$. Le digraphe G admet alors un réseau expansif pour n'importe quel multiple de $s := \prod_{i=1}^{j} p_i^{d_i}$. En effet, prenons n'importe quel entier s'. On peut le décomposer en un produit de puissances de nombres premiers $p'^{d'_1}_1 p'^{d'_2}_2 \ldots$. Le produit ss' va nous donner une multiplication de puissances de nombres premiers. La puissance d''_i de tout nombre premier

p_i avec $i < j$ dans ss' est égale à $d_i + d'_i \geq d_i$ et donc $p_i^{d''_i} \geq q$. Grâce au théorème 14, on sait qu'il existe un réseau corps fini expansif linéaire sur $\mathrm{F}(G, p_i^{d''_i})$. De manière similaire, si un nombre premier $p_i > p_j$ apparait dans q', par définition de j, le nombre premier est déjà suffisamment grand pour qu'il existe un réseau corps fini expansif linéaire sur $\mathrm{F}(G, p_i^{d'_i})$. On fait ensuite le produit cartésien de tous ces réseaux pour obtenir un réseau expansif linéaire sur $\mathrm{F}(G, ss')$. □

Conjecture 1. *Pour tout digraphe G fortement connexe et couvrable, il existe une taille d'alphabet k telle que l'ensemble $\mathrm{F}(G, q)$ contient un réseau expansif pour tout $q \geq k$.*

Un corolaire du théorème 14 est que choisir les coefficients α_k et β_k au hasard produit presque systématiquement un réseau expansif lorsque q est suffisamment grand. Encore plus frappant, nous pouvons définir une stratégie que l'on va appeler la *Random-Linear-Strategy*, pour construire des familles de réseaux expansifs. Considérons un nombre n et une puissance d'un nombre premier q donnés et considérons $\mathbb{A}$ doté des lois d'un corps fini $\mathrm{GF}(q)$. La *Random-Linear-Strategy* choisit d'abord une matrice aléatoire $M \in \mathbb{A}^{n \times n}$ dont les entrées sont toutes non nulles. Ensuite, pour tout digraphe G de taille n donné, la stratégie donne le réseau linéaire $f : x \mapsto x(M \odot M_G)$. On rappelle que $\odot$ est le produit matriciel de Hadamard (voir la sous-section 4.2.2). Le corolaire 5 ci-dessous nous donne une borne inférieure sur la probabilité que la *Random-Linear-Strategy* produise un réseau expansif $f \in \mathrm{F}(G, q)$ pour tout digraphe G de taille n. On peut voir que cette probabilité devient strictement positive quand la taille q devient suffisamment grande.

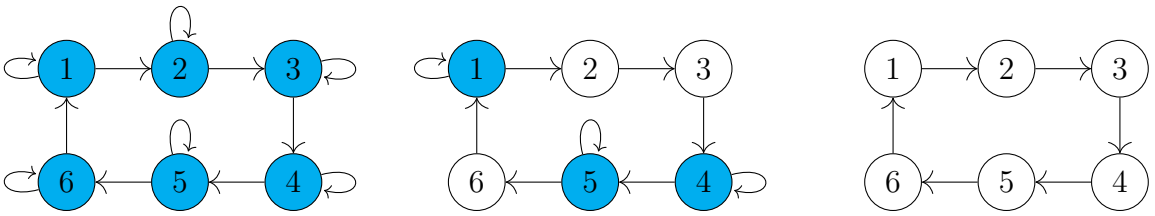
Corolaire 5. *La* Random-Linear-Strategy *a une probabilité d'au moins $1 - s/(q-1)$, avec $s = 2^{n^2-1}n^2(n+1)$ de produire un réseau expansif pour tout digraphe G fortement connexe et couvrable de taille n.*

Démonstration. Soit $\alpha = \{\alpha_{ij} \mid i, j \in [n]\}$ les variables auxquelles la *Random-Linear-Strategy* associer une valuation. En reprenant le même raisonnement que dans la démonstration du théorème 14, on peut montrer que tout digraphe G fortement connexe et couvrable de taille n, il y a au plus $\leq n \cdot n(n+1)/2 \cdot (q-1)^{n^2-1}$ valuations des variables α qui ne génèrent pas un réseau expansif sur G. De plus, il y a moins de 2^{n^2} digraphes G fortement connexes et couvrables de taille n. Par conséquent, il y a au plus $s(q-1)^{n^2-1}$ choix pour α qui ne génèrent pas un réseau expansif pour tout digraphe G valide. Donc, la probabilité de succès est d'au moins $1 - s(q-1)^{n^2-1}/(q-1)^{n^2} = 1 - s/(q-1)$. □

4.4 Graphes admettant un réseau expansif pour tout alphabet

Nous allons présenter deux familles de digraphes qui généralisent le cycle. La première famille que nous allons présenter est celle des *cycles bouclés* définie ci-dessous. Le

digraphe $G = (V, A)$ de taille n est *cycle bouclé* s'il existe un ensemble $S \subseteq V$ tel que $A = \{(i, i+1) \mid i \in V\} \cup \{(s, s) \mid s \in S\}$. On dit qu'un cycle bouclé est *propre* si $S \neq V$. Autrement dit, si le digraphe G est un cycle bouclé impropre alors il est boucle-complet. Les 3 digraphes ci-dessous sont des cycles bouclés. Celui de gauche est impropre et les deux autres sont propres. Dans les trois cas, l'ensemble S est représenté en cyan.



Nous allons maintenant présenter deux lemmes qui vont nous aider à prouver notre résultat principal sur la famille des digraphes cycle bouclé.

Lemme 19. *Pour tout digraphe G, les trois propriétés suivantes sont équivalentes :*

1. *Le réseau XOR sur G est bijectif ;*
2. *La matrice d'adjacence M_G est inversible pour le corps fini* $\mathrm{GF}(2)$ *;*
3. *G a un nombre impair de décompositions en cycles.*

Démonstration. Voyons pourquoi la propriété 2 est équivalente à la propriété 1. Le réseau d'automates XOR sur G est défini comme la fonction $f : x \mapsto xM_G$. Par définition, M_G est inversible si et seulement si elle a un inverse M'. On peut alors définir $f^{-1} : x \mapsto xM'$. On a alors $f^{-1}f(x) = f^{-1}(xM_G) = xM_GM' = x$ et f est bijective. À l'inverse, si M_G est non inversible, alors la fonction $f : x \mapsto xM_G$ est non bijective.

Maintenant, montrons l'équivalence entre la propriété 2 et la propriété 3. On sait qu'une matrice est inversible si et seulement si son déterminant est non nul. Comme on travaille sur un corps fini de taille 2 (où $-1 = 1$), on peut simplifier la formule de Leibniz (voir sous-section 4.2.2) comme suit :

$$\det(M_G) := \sum_{\pi \in \mathbb{S}_{[n]}} \prod_{i \in [n]} M_G(i, \pi(i)).$$

Il nous suffit maintenant de montrer que, pour tout $\pi \in \mathbb{S}_{[n]}$, on a $\prod_{i \in [n]} M_G(i, \pi(i)) = 0$ si et seulement si π ne correspond pas à une décomposition en cycles.

Prenons π qui a chaque sommet $i \in [n]$ associe le successeur de i dans une certaine décomposition en cycle. Pour tout $i \in [n]$, il existe un arc $(i, \pi(i)) \in A$ et donc $M_G(i, \pi(i)) = 1$. On a donc $\prod_{i \in [n]} M_G(i, \pi(i)) = 1$. À l'inverse, prenons π qui ne correspond pas à une décomposition en cycles. Il existe donc $j \in V$ avec $(j, \pi(j)) \notin A$. On a donc $M_G(i, \pi(i)) = 0$

et $\prod_{i\in[n]} M_G(i,\pi(i)) = 0$. Par conséquent, $\det(M_G)$ est équivalent au nombre de décompositions en cycle de G modulo 2. Cela signifie que $\det(M_G) = 1$ si et seulement si M_G a un nombre pair de décompositions en cycles.

□

En appliquant le même raisonnement, on peut prouver le lemme suivant.

Lemme 20. *Si un digraphe G a une unique décomposition en cycles, alors* $\det(M_G) \in \{1,-1\}$ *sur tous les anneaux $\mathbb{Z}_q$.*

Proposition 4. *Si G est un cycle bouclé alors G admet un réseau expansif linéaire pour tout $q \geq 2$, sauf si G est impropre et que $q = 2$.*

Démonstration. D'après la propriété 2 du lemme 15, pour prouver que le réseau $f : x \mapsto xM$ est expansif il suffit de prouver que la matrice M est inversible et que, pour tout $j \in [n]$, la matrice

$$N_j^{(0)} = N_j^{[0,n[} = \Big(\; M^0\big([n],j\big) \;\Big|\; M^1\big([n],j\big) \;\Big|\; \dots \;\Big|\; M^{n-1}\big([n],j\big) \; \Big),$$

est inversible également.

Supposons d'abord que G est un cycle bouclé propre et posons $M = M_G$. On peut voir que G a une unique décomposition en cycle qui est composée d'un unique cycle $(1,2,\dots,n,1)$. D'après le lemme 20, la matrice M_G est inversible. Sans perte de généralité, prenons $j = n$ et prouvons que $N_j^{(0)}$ est une matrice triangulaire supérieure avec uniquement des 1 sur sa diagonale. D'abord, pour tout $k \in [n]$, $N_j^{(0)}(n-k,n-k) = M^{n-k-1}(i,n-k) = 1$ (car le seul chemin entre $n-k$ et n de taille $n-k-1$ est le chemin de sommet $(n-k,n-k+1,\dots,n-1,j)$). De même, si on prend $k > k'$, $N_j^{(0)}(j-k,j-k') = M^{j-k-1}(j-k',j) = 0$ (car il n'y a pas de chemin entre $n-k'$ et n de taille $n-k$). Donc, $N_j^{(0)}$ est une matrice triangulaire supérieure avec uniquement des 1 sur sa diagonale. De ce fait, son déterminant égale 1 et la matrice $N_j^{(0)}$ est inversible.

Si G est un cycle bouclé impropre, on peut voir qu'il a exactement deux décompositions en cycles : la décomposition composée de l'unique cycle $(1,2,\dots,n,1)$ mais également la décomposition formée des n boucles $(1,1),\dots,(n,n)$. De ce fait, d'après le lemme 19, le réseau XOR est non bijectif et G n'admet aucun réseau linéaire pour $q = 2$.

En revanche, pour $q \geq 3$ on peut trouver un réseau bijectif. Considérons $\mathbb{A}$ comme un anneau $\mathbb{Z}_q$ (doté des lois de multiplication et d'addition classiques modulo q). Prenons $a \in \mathbb{Z}_q \setminus \{0,1\}$ inversible par multiplication. Posons $b = 1 - a$ si le nombre n est impair et

$b = a + 1$ si n est pair, et soit

$$M = \begin{pmatrix} b & a & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \\ 1 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Encore une fois, pour tout $j \in [n]$, $N_j^{(0)}$ est triangulaire supérieure et $\det(N_j^{(0)})$ est une puissance de a, ce qui montre que la matrice $N_j^{(0)}$ est inversible. □

La seconde famille que nous allons voir est celle des *cycles de cycles*. Le digraphe $G = (V, A)$ est un *cycle de cycles* s'il est composé

— d'un unique cycle C_1, ou

— d'une union de $\ell \geq 2$ cycles disjoints $C_1, \ldots, C_\ell$ reliés comme suit : pour chaque cycle C_k il y a deux sommets $i_k, j_k \in C_k$ (qui peuvent être égaux) tels que $(j_k, i_{k+1}) \in A$ (avec $k + 1 = 1$ quand $k = \ell$). Autrement dit, i_k est l'« entré » de C_k et j_k est sa « sortie ». Seuls ces k arcs (j_k, i_{k+1}) relient les k cycles.

On dit qu'un cycle de cycles est propre s'il existe un cycle C_k tel que l'arc $(j_k, i_k) \notin A$. Autrement dit, il existe un entier k tel que l'unique voisin sortant de j_k dans C_k n'est pas i_k.

Par exemple, les trois digraphes de la figure 4.1 sont des cycles de cycles.

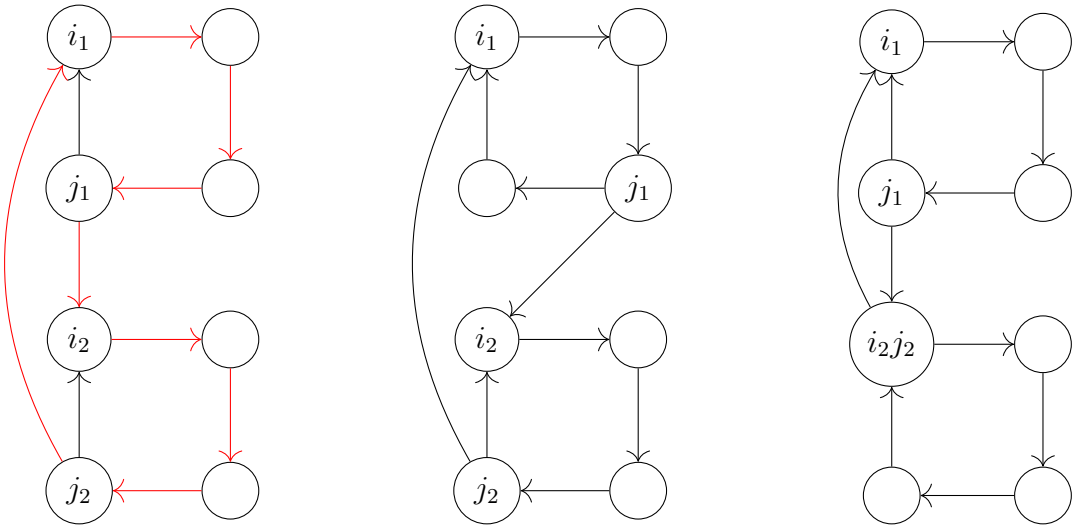


FIGURE 4.1 – Trois cycles de cycles. Celui de gauche est impropre et les deux autres sont propres. On remarque que le cycle de cycles impropre peut être recouvert entièrement par un cycle (indiqué en rouge).

Proposition 5. *Si G est un cycle de cycles, alors G admet un réseau expansif linéaire pour tout $q \geq 2$ sauf si G est un cycle de cycles impropre et que $q = 2$.*

Démonstration. Soit G un cycle de cycles propre. Premièrement, vérifions que G a une unique décomposition en cycles. C'est vrai quand G est composé d'un unique cycle. Prouvons maintenant que quand G est composé de plusieurs cycles, l'unique décomposition possible est $C_1, \ldots, C_\ell$. Soit $k \in [\ell]$, l'entier tel que $(j_k, i_k) \notin A$. Le voisin sortant j'_k de j_k n'a que j_k comme voisin entrant. Donc, il est clair que l'arc (j_k, j'_k) appartient à un cycle de la décomposition et donc que j_k et j'_k appartiennent au même cycle. Cependant, on peut voir que le seul cycle qui contient (j_k, j'_k) est C_k. En effet, le seul arc sortant de C_k est l'arc (j_k, i_{k+1}). Donc, C_k est un cycle de notre décomposition. Notons que i_k est dans C_k. Donc, le sommet j_{k-1} n'est pas dans le même cycle de la décomposition que i_k. Le sommet j_{k-1} a un unique voisin sortant j'_{k-1} qui ne soit pas dans C_k. Donc, l'arc (j_{k-1}, j'_{k-1}) appartient à un cycle de la décomposition. On en déduit que C_{k-1} est un cycle de notre décomposition également. On peut répéter ce raisonnement pour chacun des cycles jusqu'à prouver que l'unique décomposition en cycles disjoints est $C_1, \ldots, C_k$. Par conséquent, le réseau XOR sur G est bijectif.

À l'inverse, si G est un cycle de cycles impropre, on prouve qu'il a alors exactement deux décompositions en cycle. On peut voir avec le raisonnement précédent que si (j_1, i_1) appartient à un cycle d'une décomposition en cycles alors celle-ci est exactement $C_1, \ldots, C_\ell$. Maintenant si (j_1, i_1) n'appartient pas à un cycle C de la décomposition alors (j_1, i_2) doit appartenir à C. D'ici, la seule possibilité pour C est de suivre le cycle C_2 entre i_2 et j_2 (ce qui couvre tous les sommets de C_2 car $(j_2, i_2) \in A$). En poursuivant ce raisonnement, on peut vérifier que la seule possibilité est de faire un seul cycle qui contient tout V. Il y a donc exactement deux décompositions en cycles de G. Par conséquent, d'après le lemme 19, ceci signifie que le XOR sur G n'est pas bijectif.

Prenons donc $q \geq 3$. D'après le théorème 11, si G est couvrable, alors il admet un réseau linéaire bijectif $f \in \mathrm{F}(G, q)$ pour $q \geq 3$. Supposons, par l'absurde, que f n'est pas expansif. Il existe donc une configuration $x \neq (0)^n$ et un sommet $v \in [n]$ tels que, pour tout $t \geq 0$, $f_v^t(x) = 0$. Il y a deux cas à considérer.

Premièrement, supposons que, pour tout cycle $C_k = (c_1, \ldots, c_p, c_1)$ avec $c_1 = i_k$ (et donc $c_p = j_k$), il existe un temps $t \geq 0$ tel que $f_{i_k}^t(x) \neq 0$. On remarque que, pour tout $a \in [2, p]$, la fonction locale f_{c_a} est définie comme $f_{c_a} : y \mapsto \alpha_a y_{c_{a-1}}$ avec $\alpha_a = M(c_{a-1}, c_a) \neq 0$.

Pour tout $a \in [p]$, on a donc

$$f_{c_a}^{t+a-1}(x) = f_{c_1}^t(x)\alpha_1\alpha_2 \ldots \alpha_a \neq 0.$$

Par conséquent, pour tout sommet c du cycle C, on a une étape de temps t' telle que $f_c^{t'}(x) \neq 0$. En particulier, v appartient à un certain cycle C_k et il existe donc une étape de

temps t' telle que $f_v^{t'}(x) \neq 0$, ce qui est une contradiction.

Deuxièmement, supposons qu'il existe un cycle $C_k = (c_1, \ldots, c_p, c_1)$ tel que, pour tout $t \geq 0$, $f_{c_1}^t(x) = 0$. Donc, pour tout, $t \geq p$ et $a \in [p]$,

$$f_{c_a}^{t+a-1}(x) = f_{c_1}^t(x)\alpha_1\alpha_2 \ldots \alpha_a = \mathtt{0}.$$

En particulier, pour tout $t \geq p+1$, $f_{i_k}^t(x) = \mathtt{0}$ et

$$f_{i_k}^{t+1}(x) = \alpha f_{i'_k}^t(x) + \beta f_{j_{k-1}}^t(x) = \alpha 0 + \beta f_{j_{k-1}}^t(x) = \beta f_{j_{k-1}}^t(x),$$

avec i'_k le prédécesseur de i_k dans C_k, $\alpha = M(i'_k, i) \neq \mathtt{0}$ et $\beta = M(i'_k, i) \neq \mathtt{0}$. Clairement, si $f_{j_{k-1}}^t(x) \neq 0$ alors $f_{i_k}^{t+1}(x) = \beta f_{j_{k-1}}^t(x) \neq 0$, ce qui est une contradiction. Donc, pour tout $t \geq p$, $f_{j_{k-1}}^t(x) = \mathtt{0}$. Cependant, on remarque que j_{k-1} a un seul voisin j'_{k-1} et que si $f_{j_{k-1}}^{t'}(x) = \mathtt{0}$ pour tout t' supérieur à un certain t, alors $f_{j'_{k-1}}^{t'}(x)$ pour tout $t' > t+1$. On peut répéter ce raisonnement pour prouver que $f_{i_{k-1}}^{t'}(x) = 0$ pour tout t' supérieur à un certain $p+p'$ avec p' la taille de C_{k-1}. On peut répéter encore ce raisonnement pour chacun des ℓ cycles jusqu'à prouver que, pour tout $t \geq n$, $f^t(x) = (\mathtt{0})^n$. Ceci contredit le fait que le réseau f soit bijectif. Par conséquent, un réseau bijectif est forcément expansif. En outre, on remarque que l'on a affaire à une famille de graphe où être bijectif est équivalent au fait d'être expansif pour $q \geq 3$.

□

4.5 Inexistence de réseaux expansifs

Pour tout digraphe G, on écrit l'ensemble des réseaux expansifs (*resp.* abéliens expansifs) dans $\mathrm{F}(G, q)$ comme $\mathrm{E}(G, q)$ (*resp.* $\mathrm{EA}(G, q)$). Nos résultats sur l'inexistence de réseaux expansif sont basés sur la famille des graphes *étoiles*. Pour tout $n \geq 2$, l'étoile de taille n est le digraphe $G_n = (V, A)$ avec $V = [n]$ et $A = \{(1, i), (i, 1)\} \cup \{(i, i) \mid i \in [2, n]\}$.

Théorème 15. *Pour tout $q \geq 2$, il existe un digraphe fortement connexe et couvrable G tel que $\mathrm{E}(G, q) = \emptyset$.*

Démonstration. Soit $q \geq 2$ fixé. Soient $s = |\{\phi' : \mathbb{A}^2 \to \mathbb{A}\}| = q^{q^2}$, $p = q^{s+q}$, $m > (q^{2p^2})$, $\ell = \lceil \log_q(m) \rceil$ et $n \geq \ell s$. Nous allons montrer que $E(G_n, q) = \emptyset$. Considérons un réseau $f \in \mathrm{F}(G_n, q)$ et prouvons qu'il n'est pas expansif. On peut considérer que f est bijectif car s'il ne l'était pas, on saurait déjà qu'il n'est pas expansif.

Pour chacune des s fonctions $\phi' : \mathbb{A}^2 \to A$, notons $I_{\phi'}$ l'ensemble des coordonnées $i \in [2, n]$ tel que $\tilde{f}_i = \phi'$. Étant donné la valeur de n choisie $(\ell s + 1)$, il y a au moins une fonction $\phi : \mathbb{A}^2 \to \mathbb{A}$ telle que $|I_\phi| \geq \ell$. Étant donné la valeur de ℓ choisie $(\lceil \log_q(m) \rceil)$, il existe au

moins m configurations $z \in \mathbb{A}^n$ telles que $z_{[n]\setminus I_\phi} = (0)^{|[n]\setminus I_\phi|}$. On va les noter $z^{(1)}, \dots, z^{(m)}$. Comme f est bijectif, pour tout $k \in [m]$, $z^{(k)}$ fait partie d'un cycle limite de f. Notons $\ell_1, \dots, \ell_m$ la taille des cycles limites de $z^{(1)}, \dots, z^{(m)}$. Montrons que, pour tout $k \in [m]$, $\ell_k \leq p$. D'abord, prouvons par récurrence sur t que, pour tout $t \geq 0$ et $i, j \in [2, n]$, on a $f_i^t(z^{(k)}) = f_j^t(z^{(k)})$ si l'une des deux conditions suivantes est respectée : soit $\{i, j\} \subseteq I_{\phi'}$ avec $\phi' \neq \phi$, soit $\{i, j\} \in I_\phi$ et $z_i^{(k)} = z_j^{(k)}$.

C'est clairement vrai pour $t = 0$ par définition de $z^{(k)}$. Maintenant, supposons que, pour $t \geq 0$, on ait $a = f_i^t(z^{(k)}) = f_j^t(z^{(k)})$. Soit $b = f_1^t(z^{(k)})$ et $\phi' : \mathbb{A}^2 \to \mathbb{A}$ tels que $\tilde{f}_i = \tilde{f}_j = \phi'$ avec éventuellement $\phi' = \phi$. On a :

$$f_i^{t+1}(z^{(k)}) = \phi'(b, f_i^t(z^{(k)})) = \phi'(b, a) = \phi'(b, f_j^t(z^{(k)})) = f_j^{t+1}(z^{(k)}).$$

Ceci prouve l'hypothèse de récurrence.

Posons $\{I_1, \dots, I_{s-1}\} = \{I_{\phi'} \mid \phi' : \mathbb{A}^2 \to \mathbb{A}\}$ avec $I_1 = I_\phi$. Maintenant, considérons l'ensemble Z des configurations du cycle limite de $z^{(k)}$. On a donc $Z = \{f^t(z^{(k)})\}$. Analysons une configuration $z \in Z$. Comme prouvé dans la récurrence ci-dessus, pour tout $r \in [s-1]$ et $i, j \in I_r$, on a $z_i = z_j$. Donc, z_{I_r} ne peut prendre que q valeurs différentes, au mieux. De la même façon, pour tout $a \in \mathbb{A}$, on peut définir J_a comme l'ensemble des coordonnées i de I_ϕ telles que $z_i^{(k)} = a$. Clairement, z_{J_a} ne peut prendre que q valeurs différentes, au mieux, tout comme z_1. Par conséquent, Z est de taille au plus $p = q^{s+q}$ et donc $\ell_k \leq p$.

Pour tout $k \in [m]$, notons $\rho^{(t)}(z^{(k)})$ et $\rho(z^{(k)})$ les traces respectivement de taille t ou infinie sur le sommet 1. Étant donné la valeur de m choisie ($> (q^{pp})^2$) il existe $k, k' \in [m]$ tels que $\rho^{(pp)}(z^{(k)}) = \rho^{(pp)}(z^{(k')})$. On va noter $x = z^{(k)}$ et $y = z^{(k')}$. Il ne reste plus qu'à montrer que l'on a également $\rho(x) = \rho(y)$. On remarque que les traces $\rho(x)$ et $\rho(y)$ sont périodiques sur des périodes qui n'excèdent pas les tailles des cycles limites $\ell_x = \ell_k$ et $\ell_y = \ell_{k'}$. Donc, la concaténation terme à terme de $\rho(x)$ et $\rho(y)$ est périodique sur une période d'au plus $\mathrm{PPCM}(\ell_x, \ell_y) \leq \ell_x \ell_y \leq pp$. Comme $\rho^{(pp)}(x) = \rho^{(pp)}(y)$, on peut conclure que $\rho(x) = \rho(y)$ alors que $x \neq y$. Donc, f n'est pas expansif et, plus généralement, $\mathrm{E}()(G_n, q) = \emptyset$. □

On montre maintenant que la borne inférieure sur le plus petit n tel qu'il existe G couvrable et fortement connexe de taille n qui n'a pas de réseau expansif sur $\mathbb{A}$ peut être significativement baissée si on ne considère que les réseaux linéaires. On donne une preuve qui fonctionne en fait pour tous les réseaux abéliens avec une borne quasi-polynomiale.

Théorème 16. *Pour tout $q \geq 2$ et tout $n > q^{2\log(q)}$, on a* $\mathrm{EA}(G_n, q) = \emptyset$.

Démonstration. Soit n_q le nombre maximum d'endomorphismes d'un groupe abélien d'ordre q. Par le théorème de décomposition des groupes abéliens en produit de groupes cycliques, on peut voir qu'un endomorphisme est déterminé par sa valeur sur au plus $\log(q)$ éléments

(éléments égaux au générateur sur une composante du produit et 0 sur les autres). Donc, $n_q \leq q^{\log(q)}$.

On suppose que l'alphabet $\mathbb{A}$ est doté d'une loi de groupe abélien d'ordre q et soit $f \in \mathrm{F}(G_n, q)$ un endomorphisme de $\mathbb{A}^n$. Étant donné la valeur de n choisie ($n > q^{2\log(q)} \geq (n_q)^2$), il doit exister deux sommets $i \neq 1$ et $j \neq 1$ tels que $f_i(x) = e'(x_i) + h(x_1)$, $f_j(x) = e'(x_j) + h'(x_1)$ et que $f_1(x) = e(x_i) + e(x_j) + h''(x_{V\setminus\{i,j\}})$ pour un certain couple d'endomorphismes e et e' de $\mathcal{A}$. Considérons une configuration non nulles x telle que $x_i + x_j = 0$ et $x_k = 0$ pour tout $k \in V \setminus \{i, j\}$. on a alors

$$\begin{aligned} f_i(x) + f_j(x) = e'(x_i) + e'(x_j) = e'(x_i + x_j) = 0; \\ f_1(x) = e(x_i) + e(x_j) + h''(0^{n-2}) = e(x_i + x_j) = e(0) = 0; \\ \forall k \in [n] \setminus \{1, i, j\}, f_k(x) = g^{(k)}(x_k x_1) = g^{(k)}(00) = 0. \end{aligned}$$

Par récurrence, on obtient que, pour tout $t \geq 1$, $f_1^t(x) = 0$. Le réseau f est donc non expansif. □

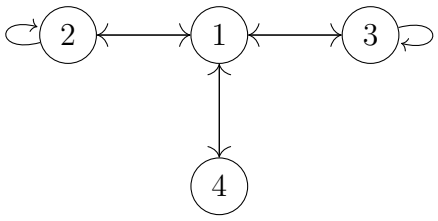
La preuve peut facilement être adaptée pour des réseaux linéaires. De ce fait, on peut trouver une borne polynomiale sur le plus petit n pour lequel G_n n'admet aucun réseau linéaire pour un alphabet de taille q.

Corolaire 6. *Le graphe G_n n'admet pas de réseau linéaire expansif pour q quand $n > (q-1)^2$.*

On conjecture qu'en fait, il y a une grande différence entre admettre un réseau expansif et admettre un réseau expansif abélien.

Conjecture 2. *Pour tout $q \geq 2$, il existe un digraphe G tel que $E(G, q) \neq \emptyset$ mais que $EA(G, q) = \emptyset$.*

Dans la proposition 6, on donne quelques arguments en faveur de la conjecture 2 en montrant qu'elle est vraie pour tout $q \equiv 2 \mod 4$. Soit G le digraphe à 4 sommets représenté dans la figure ci-dessous.



Proposition 6. *Considérons le digraphe G présenté dans la figure ci-dessus. Nous avons $\mathrm{E}(G, q) \neq \emptyset$ pour tout $q \geq 2$ et $\mathrm{EA}(G, q) \neq \emptyset$ si et seulement si $q \not\equiv 2 \mod 4$.*

Démonstration. Premièrement, vérifions que G n'admet aucun réseau abélien expansif pour $q \not\equiv 2 \mod 4$.

Soit, $f \in \mathrm{F}(G, 2)$ un réseau abélien. On sait que f est le réseau XOR $f : x \mapsto xM_G$. On peut vérifier que la configuration $x = \mathtt{0110}$ est un point fixe de f et, bien sûr, $(\mathtt{0})^n$ aussi. Donc, pour tout $t \geq 0$, $f_1^t(x) = f_1^t((\mathtt{0})^n) = \mathtt{0}$ et f n'est pas expansif.

Plus généralement, pour tout $q = 2k$ avec k impair, considérons un réseau abélien $f \in \mathrm{F}(G, 2k)$. Comme 2 et k sont premiers entre eux, on peut voir l'alphabet $[0, 2k[$ comme le produit de deux alphabets $\{0, 1\}$ et $K = [0, k[$. De ce fait, on peut trouver deux fonctions abéliennes $g \in \mathrm{F}(G, 2)$ et $h \in \mathrm{F}(G, k)$ telles que $f = g \times h$. Cependant, on a vu que g doit être un réseau XOR et donc qu'il n'est pas expansif. Donc, f n'est pas expansif non plus. Par conséquent, G n'admet pas de réseau abélien expansif quand $q \equiv 2 \mod 4$.

Deuxièmement, on montre qu'il existe un réseau abélien expansif sur G pour tout $q \not\equiv 2 \mod 4$. On va montrer que c'est vrai pour tout nombre premier q inégal à 2 et le cas général en découlera grâce au produit cartésien de fonctions. Soit $q \neq 2$ un nombre premier et soit $\alpha \neq \{0, 1\}$ un élément de $\mathrm{GF}(q)$. Soit $f : x \mapsto xM$, avec M la matrice suivante :

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & \alpha & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

On peut calculer que $\det(M) = -\alpha \neq \mathtt{0}$ et la fonction f est donc bijective. De plus, on peut calculer les déterminants de $N_0, \ldots, N_3$ comme suit :

$$N_0 = \begin{pmatrix} 1 & 0 & 3 & \alpha+1 \\ 0 & 1 & 1 & 4 \\ 0 & 1 & \alpha & \alpha^2+3 \\ 0 & 1 & 0 & 3 \end{pmatrix}, \qquad \det(N_0) = \alpha^2 - \alpha;$$

$$N_1 = \begin{pmatrix} 0 & 1 & 1 & 4 \\ 1 & 1 & 2 & 3 \\ 0 & 0 & 1 & \alpha+1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \qquad \det(N_1) = \alpha;$$

$$N_2 = \begin{pmatrix} 0 & 1 & \alpha & \alpha^2+3 \\ 0 & 0 & 1 & \alpha+1 \\ 1 & \alpha & \alpha^2+1 & \alpha^3+2\alpha \\ 0 & 0 & 1 & \alpha \end{pmatrix}, \qquad \det(N_2) = -1;$$

$$N_3 = \begin{pmatrix} 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & \alpha \\ 1 & 0 & 1 & 0 \end{pmatrix}, \qquad \det(N_3) = 1 - \alpha.$$

Tous les déterminants sont non nuls. Donc, f est expansif.

Troisièmement, pour $q = 2$, on peut vérifier par ordinateur que le réseau f (non abélien) dont les fonctions locales sont données ci-dessous.

$$\begin{aligned} f_0(x) &= x_1x_2 + x_3 + 1, \\ f_1(x) &= x_0 + x_1, \\ f_2(x) &= x_0 + x_2 + 1, \\ f_3(x) &= x_0. \end{aligned}$$

En utilisant notre résultat précédent qui dit que G admet un réseau expansif pour tout alphabet non paire et le produit cartésien, on en déduit que G admet un réseau expansif pour tout $q \geq 2$.

□

4.6 Temps d'expansion

Considérons un réseau expansif $f \in \mathrm{F}(n, q)$. Comme le réseau f est bijectif, pour toute coordonnée i et toute configuration $x \in \mathbb{A}^n$, la trace $\rho_i(x)$ est périodique. Notons $C_x = \{f(x), \ldots, f^{\ell_x}(x) = x\}$ le cycle limite auquel appartient x. La taille ℓ de la période de la trace de $\rho_i(x)$ est au plus ℓ_x. En effet, sinon, x et $f^{\ell}(x)$ seraient deux configurations distinctes. Pour tout réseau $f \in \mathrm{F}(n, q)$, $\{x, y\} \subseteq \mathbb{A}^n$ et $i \in [n]$, soit

$$\tau_i(x, y) = \min\left\{t \geq 1 \mid f_i^t(x) \neq f_i^t(y)\right\}.$$

De manière équivalente, $\tau_i(x, y) = t + 1$ avec $t \geq 0$ le plus grand entier tel que $\rho_i^t(x) = \rho_i^t(y)$. Le *temps d'expansion* de f est alors

$$\mathfrak{t}(f) := \max\{\tau_i(x, y) \mid \{x, y\} \subseteq \mathbb{A}^n, i \in [n]\}.$$

Autrement dit, c'est le plus court temps pour lequel l'évolution de x_i détermine totalement x pour tout x et pour tout $i \in [n]$. Pour tout $i \in [n]$ et $t \geq 0$, il est clair que si la fonction $\rho_i : \mathbb{A}^n \to \mathbb{A}^t$ est injective, alors $t \geq n$. On en conclut que, pour tout $f \in F(n, q)$, $\mathfrak{t}(f) \geq n$.

On dit que f est fortement expansif s'il est expansif et que $\mathfrak{t}(f) = n$. Le lemme 15 montre que tout réseau expansif et linéaire sur un corps fini est fortement expansif. Un réseau fortement expansif peut être vu comme suit. Pour tout $x \in \mathbb{A}^n$, considérons la

matrice

$$M_x = \begin{pmatrix} f_1(x) & f_2(x) & \dots & f_n(x) \\ f_1^2(x) & f_2^2(x) & \dots & f_n^2(x) \\ \vdots & \vdots & \dots & \vdots \\ f_1^n(x) & f_2^n(x) & \dots & f_n^n(x) \end{pmatrix}.$$

Clairement, f est bijective si et seulement si on peut déduire x depuis toute ligne de M_x. Maintenant, f est fortement expansif si et seulement si on peut déduire x depuis toute colonne de M_x.

On rappelle que le temps d'expansion est la valeur maximum de t tel que l'on puisse retrouver tout $x \in \mathbb{A}^n$ depuis les t premières valeurs de sa trace sur n'importe quelle coordonnée $i \in [n]$. Toutefois, pour certaines valeurs de x et de i particulières, ce temps peut être plus petit que n comme montré dans l'exemple suivant.

Exemple 1. Considérons le réseau $f \in \mathrm{F}(3,2)$ défini comme suit :

x	$f(x)$
000	001
001	110
010	101
011	111
100	011
101	010
110	000
111	100

.

On peut vérifier que le réseau f est en effet expansif, avec un temps d'expansion $\iota(f) = 4$ (et donc f n'est pas fortement expansif). Alors la trace sur la coordonnée $i = 1$ est comme suit (la partie de la trace qui suffit à retrouver l'état initial est présentée en gras) :

x	$(f_1(x), f_1^2(x), f_1^3(x), f_1^4(x))$
000	**0100**
001	**100**1
010	**1010**
011	**11**01
100	**011**0
101	**0101**
110	**00**10
111	**1011**

.

En particulier, les configurations 011 et 110 peuvent être retrouvées après seulement deux étapes de temps.

En revanche, on montre dans la proposition 7 ci-dessous que le temps d'expansion d'un réseau fortement expansif est « universel ». En d'autres termes, on montre que, pour tout $i \in [n]$ et $x \in \mathbb{A}^n$, on doit attendre n étapes avant d'être capable de retrouver x.

Proposition 7. *Si le réseau $f \in \mathrm{F}(n,q)$ est fortement expansif alors, pour tout $i \in [n]$ et $x \in \mathbb{A}^n$, il existe une configuration $y \neq x$ telle que $\tau_i(x,y) = n$.*

Démonstration. Si le réseau $f \in \mathrm{F}(n,q)$ est fortement expansif, alors pour tout $i \in [n]$, la fonction $\rho_i^{(n)}$ est surjective. Supposons par l'absurde qu'il existe une coordonnée $i \in [n]$ et une configuration $x \in \mathbb{A}^n$ telles que, pour toute configuration $y \in \mathbb{A}^n \setminus \{x\}$, on ait $\rho_i^{(n-1)}(x) \neq \rho_i^{(n-1)}(y)$.

Soit $a \neq f_i^n(x)$, alors il n'y a pas de configuration $z \in \mathbb{A}^n$ telle que $\rho_i^{(n)}(z) = \rho_i^{(n-1)}(x)a$. Ceci contredit la surjectivité de $\rho_i^{(n)} : \mathbb{A}^n \to \mathbb{A}^n$ et donc la forte expansivité. □

Pour souligner la spécificité des réseaux fortement expansifs, nous allons montrer dans le théorème 14 que le temps d'expansion maximum que peut atteindre un réseau expansif $f \in F(n,q)$ est presque q^n.

Théorème 17. *Pour n et q, le maximum $t(f)$ atteint par un réseau $f \in \mathrm{F}(n,q)$ est entre $q^n - q - 1$ et $q^n - 2$.*

Démonstration. Borne supérieure. Soit $\{x,y\} \subseteq \mathbb{A}^n$ avec $\ell_x \leq \ell_y$ et $i \in [n]$ tels que $t(f) = \tau_i(x,y)$. Notons $\tau = \tau_i(x,y)$.

Cas 1 : $\ell_x \mid \ell_y$ (*i.e.* ℓ_x divise ℓ_y). On prouve d'abord que $\tau < \ell_y$. Supposons par l'absurde que ce ne soit pas le cas. Par conséquent, pour tout $t \in [\ell_y]$, on a $f_i^t(x) = f_i^t(y)$. Alors, $\rho_i(x) = \rho_i(y)$, ce qui contredit l'expansivité de f. On a donc bien $\tau < \ell_y \leq q^n$ et donc $\tau \leq q^n - 1$. Supposons par l'absurde que $\tau = q^n - 1$. On a donc $\ell_y = q^n$. Donc, le réseau f est une permutation cyclique de $\mathbb{A}^n$ et la trace $\rho_i(x)$ est une permutation cyclique de $\rho_i(y)$. De plus, la seule différence entre les deux traces est à la position q^n. En d'autres termes, on a $f^{q^n}(x) = x$, $f^{q^n}(y) = y$ et $x_i \neq y_i$. Soit $a = x_i$ et $b = y_i$. Soit $s = |\{t \in [q^n - 1] \mid f_i^t(x) = a\}|$ le nombre de fois que la trace de x égale a avant l'étape de temps q^n. Comme il y a q^{n-1} configurations z telles que $z_i = a$ et $f_i^{q^n}(x) = x_i = a$, on a $s = q^{n-1} - 1$. Toutefois, s correspond aussi au nombre de fois ou y égale a avant l'étape de temps $q^n - 1$ (parce que l'on a supposé par l'absurde que $\tau = q^n - 1$ et donc que $f_i^t(x) = f_i^t(y)$ pour tout $t \in [q^n - 1]$). On obtient $s = q^{n-1}$, ce qui est la contradiction voulue. Donc, $\tau \leq q^n - 2$.

Cas 2 : $\ell_x \nmid \ell_y$ (*i.e.* ℓ_x ne divise pas ℓ_y). Prouvons que $\tau \leq \ell_x + \ell_y - \mathrm{PGCD}(\ell_x, \ell_y) - 1$. Supposons par l'absurde que $\tau \geq \ell_x + \ell_y - \mathrm{PGCD}(\ell_x, \ell_y)$. On va découper nos traces en $\mathrm{PGCD}(\ell_x, \ell_y)$ blocs. Disons que le premier bloc de la trace de x est $X = (u_1, \ldots, u_{|X|})$ et que celui de y est $Y = (u_1, \ldots, u_{|Y|})$ (c'est cohérent car X est un préfixe de Y). On a donc $|X| = \ell_x/\mathrm{PGCD}(\ell_x, \ell_y)$ et $|Y| = \ell_y/\mathrm{PGCD}(\ell_{x,y})$. Par définition du PGCD, $|X|$ et $|Y|$ sont

premiers entre eux. Soit α et a respectivement le quotient et le reste de la division de $|Y|$ par $|X|$. Soit β et b respectivement le quotient et le reste de la division de $|X|$ par a. Autrement dit, on a $|Y| = \alpha|X| + a$ avec $a \in [0, |X|[$ et $|X| = \beta a + b$ avec $b \in [0, a[$. On voit que a et b sont premiers entre eux.

Affirmation 6. *Soit $u = u_1 \dots u_a$ et $u' = u_1 \dots u_b$. Alors, $X = u^\beta u'$ et $Y = X^\alpha u$.*

Démonstration. Clairement, $Y = X^\alpha v$ avec $v = u_{\alpha|X|+1} \dots u_{|Y|}$. Aux étapes de temps entre $\alpha|X| + 1$ et $\alpha|X| + a$, la trace de x décrit u, et donc $v = u$. Ceci montre la seconde partie de l'affirmation. De façon similaire, aux étapes de temps entre $|Y| + 1 = \alpha|X| + a + 1$ et $|Y| + a = \alpha|X| + 2a$, la trace de y correspond à u, donc X commence par u, u. Par une simple induction, on montre que u est répété tout le long de X et on obtient $X = u^\beta u'$.

■

Nous allons maintenant étudier les étapes de temps entre $t := |X| + |Y| - a - b + 1$ et $t + a + b - 1 = |X| + |Y|$. La trace de x est $u_1 \dots u_b u_1 \dots u_{a-1}$, et la trace de y est $u_1 \dots u_a u_1 \dots u_{b-1}$. Pour les étapes de temps entre $t + b + 1$ et $t + a$, on a $u_i = u_{i+b}$ pour tout $i \in [a - b]$. Pour les étapes de temps entre $t + a + 1$ et $t + a + b - 1$, on a $u_j = u_{j-a+b}$ pour tout $j \in [a - b + 1, a - 1]$. Comme b est premier avec a, il est facile de vérifier que $u_1 = \dots = u_a$. Donc, $X = u_1 \dots u_1$, ce qui contredit sa périodicité.

Borne inférieure. On construit maintenant un réseau expansif $f \in \mathrm{F}(n, q)$ tel que $\mathfrak{t}(f) \geq q^n - q - 1$. Pour donner une intuition, ce réseau est la fonction succession d'une énumération de $\mathbb{A}^n$ particulière, qui peut être vue comme un « faux ordre lexicographique ». Plus formellement, pour tout $a \in [0, q^n - 1]$, posons $a = a_0 + a_1 q + \dots + a_n q^{n-1}$. Soit $x^{(a)} = (x_1^{(a)}, \dots, x_n^{(a)}) \in \mathbb{A}^n$ définies comme $x_n^{(a)} = a_n$ et pour tout $i \in [n - 1]$,

$$x_i^{(a)} = \begin{cases} q-2 & \text{si } a_{i+1} = \dots = a_n = q-1 \text{ et } a_i = q-1; \\ q-1 & \text{si } a_{i+1} = \dots = a_n = q-1 \text{ et } a_i = q-2; \\ a_i & \text{sinon.} \end{cases}$$

Alors, soit $f : x^{(a)} \mapsto x^{(a+1)}$. Clairement, le réseau f est bijectif car la fonction $a \mapsto x_n^{(a)}$ est bijective.

On prouve maintenant que f est expansif. On a seulement besoin de montrer que, pour tout $b \in [\lfloor q^n/2 \rfloor]$ et tout $i \in [n]$, il existe un temps $t \in [0, q^n - 1]$ tel que $x_i^{(t)} \neq x_i^{(b+t)}$. Soit k le plus grand nombre tel que q^{k-1} divise b. Pour tout $i \in [k]$, on a $x_i^{(q^n-1-b)} = q - 1$ et $x_i^{(q^n-1)} = q - 2$, et pour $i \in [k + 1, n]$, on a $x_i^{(q^{i-1}-1)} = 0$ et $x_i^{(q^{i-1}-1+b)} \neq 0$. Finalement, il est facile de vérifier que $\tau_1(x^{(q^n-1)}, x^{(q-1)}) = q^n - q - 1$. □

4.7 La super-expansivité

La notion d'expansivité vue jusque-là consiste à pouvoir déterminer une configuration initiale en observant la trace de n'importe quelle coordonnée. Dans cette section, on va étudier une notion plus forte où on veut pouvoir retrouver la configuration initiale depuis n'importe quelle « observation » du réseau suffisamment grande sur les n premières étapes. Soit $f \in \mathrm{F}(n,q)$. Considérons un ensemble d'observations $\omega = \{\omega_1, \dots, \omega_n\} \subseteq [n] \times [n]$ avec, pour tout $i \in [n]$, $\omega_i = (s_i, t_i)$, s_i représentant un sommet et t_i une étape de temps. *La trace d'observation* associée est la fonction $\tau_\omega : \mathbb{A}^n \to \mathbb{A}^n$ telle que

$$\tau_\omega(x) = \Big(f_{s_1}^{t_1}(x), f_{s_2}^{t_2}(x), \dots, f_{s_n}^{t_n}(x)\Big).$$

On dit que f est *super-expansif* si pour tout ω, la fonction τ_ω est injective. De manière équivalente, le réseau f est super-expansif si on peut retrouver x en observant n'importe quel ensemble de n entrées de la matrice M_x définie précédemment.

Proposition 8. *Soit G le digraphe de taille n. Si le réseau $f \in \mathrm{F}(G,q)$ est super-expansif, alors G est le digraphe complet (i.e. il a n^2 arcs).*

Démonstration. Supposons que G n'ait pas d'arc $(i,j) \in [n]^2$ et considérons le réseau $f \in \mathrm{F}(G,q)$. Soient x et $y \in \mathbb{A}^n$ avec $x_i \neq y_i$ et $y_{[n]\setminus\{i\}} = x_{[n]\setminus\{i\}}$. Prenons également les configurations x' et y' telles que $f(x') = x$ et $f(y') = y$.

Considérons l'observation $\omega = \{\omega_1, \dots, \omega_n\} \subseteq [n] \times [n]$ telle que $\omega_i = (j, 2)$ et, pour tout $k \in [n] \setminus \{i\}$, $\omega_k = (k, 1)$. On a

$$\tau_\omega(x') = x_j f_{[n]\setminus\{i\}}(x) = y_j f_{[n]\setminus\{i\}}(y) = \tau_\omega(y').$$

Le réseau f n'est pas super-expansif. □

En utilisant une technique similaire à celle utilisée dans la preuve du théorème 14, on peut montrer l'existence de réseaux super-expansifs.

Théorème 18. *Pour tout n et toute puissance de nombre premier $q > n^2\binom{n^2}{n}$, il existe un réseau $f \in \mathrm{F}(n,q)$ super-expansif linéaire sur un corps fini* $\mathrm{GF}(q)$.

Démonstration. La technique de preuve est similaire à celle du théorème 14. Premièrement, considérons un réseau linéaire $f \in \mathrm{F}(n,q)$ avec $f : x \mapsto xM$ et considérons un ensemble d'observations $\omega = \{\big(s_1, t_1\big), \dots, (s_n, t_n)\}$ avec $t_1 \leq t_2 \leq \cdots \leq t_n$. Définissons la matrice $N_\omega \in \mathbb{A}^{n\times n}$ suivante :

$$N_\omega := \Big(\ M^{t_1}\big([n], s_1\big) \ \Big|\ M^{t_2}\big([n], s_2\big) \ \Big|\ \dots \ \Big|\ M^{t_n}\big([n], s_n\big)\ \Big).$$

Par une adaptation directe du lemme 15, on peut voir que l'observation τ_ω est injective si et seulement si la matrice N_ω est inversible. Comme dans le théorème 14, la preuve de la présente démonstration n'est pas constructive. On considère les coefficients non nuls $M(i,j)$ de la matrice M comme des variables qu'on note $x((i,j))$. On peut associer chaque chemin d'arcs $w = w_1 \dots w_p$ dans le digraphe G à un monôme $X(w) := x(w_1) \dots x(w_p)$. Alors, le déterminant de N_ω est un polynôme de ces variables. En utilisant le lemme de Schwartz-Zippel, on peut montrer que si le corps fini est suffisamment grand, alors on peut toujours considérer des valuations des variables telles que $\det(N_\omega) \neq 0$ pour peu que N_ω ne soit pas le polynôme nul. En utilisant la correspondance entre les chemins sur le graphe complet et les monômes, le déterminant de la matrice N_ω peut être exprimé comme suit :

$$\det(N_\omega) = \sum_{\pi \in \mathbb{S}_{[n]}} \varepsilon(\pi) P_\pi,$$

avec chaque monôme qui apparait dans P_π sous la forme

$$\prod_{i=1}^{n} \prod_{k=1}^{t_i} X(w^{(i)}(k)),$$

avec $w^{(i)}(1) \cdots w^{(i)}(t_i)$ un chemin de taille t_i entre les sommets $\pi(i)$ et s_i. On choisit maintenant une permutation particulière π et un monôme particulier qui apparait dans P_π et nous montrons qu'il n'apparait dans aucun autre polynôme $P_{\pi'}$ avec $\pi \neq \pi'$. Notons $S = \{s_1, \dots, s_n\} \subseteq [n]$ l'ensemble des sommets observés dans ω. On rappelle que les sommets $s_1, \dots, s_n$ ne sont pas nécessairement distincts : un même sommet peut être observé plusieurs fois (mais à différentes étapes de temps). De ce fait, S n'est pas nécessairement égale à $[n]$. Considérons les n coordonnées distinctes $s'_1, \dots, s'_n \in [n]$ telles que

$$s'_i = \begin{cases} s_i & \text{si } i = \min\{k \mid s_i = s_k\}, \\ \notin S & \text{sinon.} \end{cases}$$

Remarquons que $\{s'_1, \dots, s'_n\} = [n]$. On considère la permutation $\pi : i \mapsto s'_i$, et le monôme $Y = \prod_{i \in [n]} x\Big(s'_i s'_i\Big)^{t_i - 1} x\Big(s'_i s_i\Big)$ qui apparait dans P_π. Considérons une permutation $\pi' \in \mathbb{S}_{[n]}$ quelconque et supposons que Y apparait dans $P_{\pi'}$. En d'autres termes,

$$Y = \prod_{i=1}^{n} \prod_{k=1}^{t_i} X(w^{(i)}(k)),$$

avec $w_1^{(i)} \cdots w_{t_i}^{(i)}$ un chemin de taille t_i depuis le sommet $\pi'(i)$ jusqu'au sommet s_i. On veut maintenant prouver que $\pi' = \pi$. Pour cela, prenons $s \in S$ quelconque et prouvons que $\pi(s) = \pi'(s)$. Posons $I_s = \{i \mid s_i = s\}$. Si I_s est un singleton $\{i\}$ alors $\pi'(i) = s_i$ car le seul arc entrant de s_i apparaissant dans Y est (s'_i, s_i). Sinon, soit $k = \max\Big(I_s\Big)$. On sait que $s'_k \notin S$ (car $k \neq \min\{k \mid s_i = s_k\}$), que le seul arc entrant de s'_k apparaissant dans

M est la boucle (s'_k, s'_k) et que le seul arc sortant de s'_k apparaissant dans Y est (s'_k, s). Par conséquent, le seul chemin qui contient l'arc (s'_k, s'_k) est un chemin de s'_k vers s et de taille t_k (pour atteindre la même puissance de $x\Big((s'_k, s'_k)\Big)$ que dans Y. Par conséquent, on a $\pi'(k) = s'_k$. Avec le même raisonnement on prouve que π et π' sont égaux sur I_s pour tout $s \in S$, ce qui signifie que $\pi' = \pi$. Ceci montre que $\det(N_\omega) \neq 0$.

Le degré est N_ω est clairement au plus n^2. D'après le lemme de Schwartz-Zippel (voir le théorème 7.1.4 dans le livre [108]), il y a au plus $n^2 q^{n^2-1}$ valuations des variables $x(a)$ avec $a \in A$ pour laquelle $\det(N_\omega) = 0$. Comme il y a $\binom{n^2}{n}$ valeurs de ω possibles, il y a au plus $n^2\binom{n^2}{n}q^{n^2-1}$ valuations des variables $x(a)$ avec $a \in A$ pour lesquels τ_ω n'est pas injective. Comme $q > n^2\binom{n^2}{n}$, on a $q^{n^2} > n^2\binom{n^2}{n}q^{n^2-1}$ et il existe donc un choix de valuation des variables $x(a)$ avec $a \in A$ qui nous donne un réseau est super-expansif linéaire. □

Comme application, montrons que tout réseau super-expansif (linéaire) nous donne naturellement un tableau orthogonal (linéaire) et un *code parfait*. On va maintenant définir ces deux notions (voir les livres [79] et [103] pour un aperçu du sujet).

Un s-(q, m, λ) *tableau orthogonal* (*orthogonal array* en anglais) est un tableau $\mathcal{T} \in \mathbb{A}^{n' \times m}$ avec $n' = \lambda q^s$ (et $|\mathbb{A}| = q$) et avec la propriété suivante : pour tout sous-tableau C composé de s colonnes de $\mathcal{T}$, tout mot $\mathbb{A}^s$ apparait dans exactement λ lignes de C. On appelle s la *force* de $\mathcal{T}$ et λ est son *indice*.

Quand q est un nombre premier, on dit que le tableau orthogonal $\mathcal{T}$ est linéaire si l'ensemble de ses lignes est un $\mathrm{GF}(q)$-espace vectoriel.

Un code $\mathcal{C}$ est un ensemble de mots de $\mathbb{A}^n$ et sa distance minimum $d_{\min}(\mathcal{C})$ est la distance de Hamming minimum atteinte entre deux éléments distincts de $\mathcal{C}$. Quand q est un nombre premier, un code $\mathcal{C}$ est linéaire s'il forme un sous-espace vectoriel de $\mathrm{GF}(q)^n$. Un code parfait (appelé MDS, pour *maximum distance séparable* en anglais) est un code qui atteint la *borne de Singleton* (voir le livre [103]). En d'autres termes, il respecte l'égalité, $|\mathcal{C}| = q^{n-d_{\min}(\mathcal{C})+1}$.

Le lien entre ces objets combinatoires et les réseaux super-expansif peut être décrit comme suit. Considérons une fonction $f \in \mathrm{F}(n, q)$. Soit $\mathcal{T}_f \in \mathbb{A}^{q^n \times n^2}$ le tableau tel que chaque configuration $x \in \mathbb{A}^n$ correspond à une ligne L_x avec

$$L_x = (f_1(x), f_2(x), \ldots, f_n(x), f_1^2(x), \ldots, f_n^2(x), \ldots, f_1^n(x), \ldots, f_n^n(x)).$$

Proposition 9. *Si $f \in \mathrm{F}(n, q)$ est super-expansif alors $\mathcal{T}_f$ est un tableau orthogonal de force n et d'indice 1, et son ensemble de lignes est un code parfait de distance minimum $n^2 - n + 1$. Si en plus q est un nombre premier et f est linéaire, alors $\mathcal{T}_f$ est un tableau orthogonal linéaire et son ensemble de ligne est un code parfait linéaire.*

Démonstration. Chaque ensemble de n colonnes de $\mathcal{T}_f$ est naturellement associé à un

ensemble d'observations $\omega = \{(v_1, t_1), \ldots, (v_n, t_n)\}$. Le fait que τ_ω soit injectif (car f est super expansif) veut exactement dire qu'aucune paire de lignes L_x, L_y ne peut être identique sur cet ensemble de colonne. Par conséquent, $\mathcal{T}_f$ est un tableau orthogonal de force n. Le fait que cela corresponde également à un code parfait est déjà bien connu (voir le théorème 4.21 de [79]). Pour finir, il est immédiat de voir que quand f est linéaire alors $\mathcal{T}_f$ l'est également ainsi que le code correspondant. □

De cette proposition, on peut déduire une borne inférieure sur la taille de l'alphabet d'un réseau super-expansif.

Corolaire 7. *Il n'y a pas de réseau super-expansif de taille n sur l'alphabet $\mathbb{A}$ si $q \leq n^2 - n$.*

Démonstration. La borne de *Bush* (voir le théorème page 431 de l'article [36]), affirme qu'un tableau orthogonal de taille $n' \times m$ d'indice 1, de force t sur un alphabet $\mathbb{A}$ vérifie l'inégalité $n' \leq t + q - 1$. Prenons $n' = n^2$, $m = q^2$ et $t = n$. D'après la proposition 9, pour tout $f \in \mathrm{F}(n, q)$ super expansif, $\mathcal{T}_f$ est un tableau orthogonal de taille $n^2 \times q^n = n' \times m$, d'indice 1, de force $t = n$ sur l'alphabet $\mathbb{A}$. On a donc $n' \leq t + q - 1$, $n^2 - n + 1 \leq q$. Donc, $q > n^2 - n$ sinon f ne peut pas être expansif. □

4.8 Fréquence d'expansivité

Dans la section précédente, on a montré que, pour tout réseau, on peut avoir à attendre n étapes pour différencier deux configurations distinctes x et y en observant leurs traces sur un certain $i \in [n]$. On va voir dans cette section, que cette différenciation peut se produire de manière fréquente après sa première occurrence. Nous allons tout d'abord définir une mesure de la fréquence de cette différenciation composante par composante. Pour tout $\{x, y\} \subseteq \mathbb{A}^n$ et $i \in [n]$, soit

$$\phi_i(x, y) := \frac{d_{\mathrm{H}}(\rho_i(x)^{(\ell_x \ell_y)}, \rho(y)^{(\ell_x \ell_y)})}{\ell_x \ell_y},$$

avec d_{H} la distance de Hamming (*i.e.* $d_{\mathrm{H}}(x, y) = |\{i \in [n] \mid x_i \neq y_i\}|$). On définit alors la *fréquence d'expansivité* de f comme

$$\Phi(f) = \min\{\phi_i(x, y) \mid \{x, y\} \subseteq \mathbb{A}^n, i \in [n]\}.$$

Clairement $\frac{1}{t(f)} \leq \Phi(f) \leq 1$. Toutefois, on montre dans le théorème 19 ci-dessous qu'on peut trouver des réseaux expansifs f tels que $\Phi(f)$ est aussi proche de 1 que voulu.

Théorème 19. *Pour tout réseau expansif $f \in \mathrm{F}(n, q)$, $\Phi(f) \leq \frac{q^n - q}{q^n - 1}$. De plus, pour tout q premier et pour tout n, on peut trouver un réseau $f \in \mathrm{F}(n, q)$ tel que $\Phi(f) = \frac{q^n - q}{q^n - 1}$.*

Démonstration. Borne supérieure. Soit ℓ le produit de toutes les tailles de cycles limites de f. Considérons le code $c = \{\rho_i^\ell(x) \mid x \in \mathbb{A}^n\}$ qui appartient à $\mathbb{A}^\ell$. Comme $t(f) \leq \ell$, toutes les traces sont différentes et donc $|c| = q^n$.

Affirmation 7. *La distance $d_{\min}(c)$ respecte l'inégalité $d_{\min}(c) \leq \delta := \frac{(q-1)\ell q^{n-1}}{q^n-1}$.*

> *Démonstration.* La généralisation de la borne de Plotkin par Berlekamp (voir l'équation 13.41 du livre [18]) montre que, pour tout code c de taille ℓ sur $\mathbb{A}$ et avec une distance minimum d, on a $|c| \leq \frac{dq}{dq-\ell(q-1)}$ si le dénominateur de cette fraction est positif. Comme $\delta q > \ell(q-1)$, on peut utiliser ce résultat. De plus, si on avait $d_{\min}(c) = d > \delta$, alors $|c| \leq \frac{dq}{dq-\ell(q-1)} < \frac{\delta q}{\delta q-\ell(q-1)} = q^n$, ce qui est absurde. Donc, $d \leq \delta$.
>
> ■

Par définition, il existe une paire de configurations $\{x, y\} \in \mathbb{A}^n$ telle que $d_{\mathrm{H}}(\rho_i^{(\ell)}(x), \rho_i^{(\ell)}(y)) = d_{\min}(C)$. On obtient $\phi_i(x, y) \leq \frac{\delta}{\ell} = \frac{(q-1)q^{n-1}}{q^n-1}$.

Borne inférieure Considérons l'image de la fonction $\xi \mapsto \alpha\xi$ dans $\mathrm{GF}(q)^n$, avec α un élément primitif de ce corps fini. C'est une fonction linéaire de $\mathrm{F}(n, q)$ avec $(0)^n$ comme unique point fixe. Pour tout configuration $x \in \mathbb{A}^n$, l'orbite de x contient chacun des $q^n - 1$ éléments non nuls de $\mathrm{GF}(q)^n$. Donc, pour tout $x \neq (0)^n$ et i, $\phi_i(x, (0)^n) = \frac{(q-1)q^{n-1}}{q^n-1}$. □

Dans l'autre sens, dans la construction de la preuve du théorème 17, on a construit un réseau dont la fréquence d'expansion est $2/q^n$.

4.9 Liens avec l'expansivité dans les réseaux d'automates cellulaires

Un système dynamique topologique [98] est un couple (F, X), avec X un espace métrique compact de distance d et F une fonction continue. On dit que F est expansive s'il existe un réel constant $\epsilon > 0$ tel que

$$\forall x, y \in X, x \neq y \Rightarrow \exists t \mid |\Delta(F^t(x), F^t(y))| \geq \epsilon.$$

Dans le cas général, le temps t est un entier positif et on parle d'*expansivité positive*. Si F est bijective, alors t peut être choisi négativement (c'est la formulation adoptée dans [88]).

Un automate cellulaire (à une dimension) est un système dynamique topologique $(F, \mathbb{A}^{\mathbb{Z}})$ avec $\mathbb{A}$ un alphabet fini et F une fonction définie à l'aide d'une règle locale f :

$\mathbb{A}^{2r+1} \to \mathbb{A}$ avec $r \geq 0$. Plus précisément,

$$\forall x \in \mathbb{A}^{\mathbb{Z}}, \forall i \in \mathbb{Z}, F_i(x) = f(x_{V(i)}) \text{ avec } V(i) = [i-r, i+r].$$

F est positivement expansive si et seulement si, pour tout $i \in \mathbb{Z}$, la trace suivante est bijective [98, proposition 5.48] :

$$x \mapsto (x_{V(i)}, F(x)_{V(i)}, F^2(x)_{V(i)}, \ldots),$$

ou de manière équivalente, si et seulement si pour toute paire de configuration $\{x, y\} \subseteq X$, il existe un temps $t \geq 0$ tel que $F^t(x)_{V(i)} \neq F^t(y)_{V(i)}$.

L'expansivité (positive ou non) dans les automates cellulaires a reçu beaucoup d'attention [19, 109, 121] et est encore une direction de recherche active [83], l'un des problèmes principaux étant la décidabilité de cette propriété (voir le problème 19 de [21] ou le problème 7 de [87]).

On va supposer ici que la fonction locale $f : \mathbb{A}^{2r+1} \to \mathbb{A}$ qui définit l'automate cellulaire est *totale* : le résultat dépend effectivement de chacune de ses coordonnées. L'expansivité des automates cellulaires et des réseaux d'automates peuvent être reliés de deux façons :

1. un automate cellulaire F peut être vu comme un réseau d'automates sur un graphe d'interaction infini $(\mathbb{Z}, A)$ avec $(i, j) \notin A$ si $|i - j| > r$ (avec r l'entier tel que $2r + 1$ est la taille du voisinage). Ce graphe est toujours fortement connexe et couvrable (car il contient toutes les boucles) et F comme automate cellulaire est positivement expansif si et seulement s'il est quasi-expansif en tant que réseau d'automates infini (voir le théorème 13 et les définitions précédentes) ;
2. on peut aussi restreindre un automate cellulaire F à ses configurations périodiques de période n. Dans ce cas, il peut être vu comme un réseau d'automates $f \in \mathrm{F}(n, q)$ sur le graphe d'interaction fini $(\mathbb{Z}/n\mathbb{Z}, E)$ où $(i, j) \in E$ avec $i < j$ si et seulement si $|i - j| \leq r$ ou $|i - j| \geq n - r$. Si F est un automate cellulaire positivement expansif, alors pour tout n, le réseau d'automates $f \in \mathrm{F}(n, q)$ est quasi-expansif. En revanche, l'implication inverse n'est pas forcément vraie. Par exemple, l'automate cellulaire qui fait une translation sur la gauche $F(x)_{i+1} = x_i$ n'est pas positivement expansif alors que toutes ses restrictions de tailles finies sont quasi-expansives.

4.10 Conclusion et perspectives

Dans ce chapitre, nous avons caractérisé les graphes d'interaction qui admettent des réseaux expansifs : ce sont les digraphes fortement connexes et couvrables. Un problème qui découle naturellement des travaux réalisés ici est le suivant : étant donné un graphe

d'interaction G fortement connexe et couvrable, quelle est la plus petite taille d'alphabet $\beta(G) := q$ telle que $\mathrm{F}(G, q)$ contient un réseau expansif ?

On a montré dans le théorème 14 que tout graphe d'interaction G fortement connexe et couvrable admettait un réseau expansif linéaire sur un alphabet cubique en fonction de la taille du graphe. Ceci nous donne une borne supérieure sur $\beta(G)$. À l'inverse, le théorème 15 nous apprend que l'on peut trouver des graphes G de taille n tels que $\beta(G) \geq o(\log^3(n))$.

On peut constater que l'intervalle entre ces deux bornes est très important. Cela s'explique probablement par la façon dont nous avons obtenu notre borne supérieure. En effet, premièrement, notre démonstration ne concerne que les fonctions corps fini linéaires (qui représentent une minorité des fonctions admises par le graphe) et ensuite notre preuve est non constructive. Il serait intéressant d'avoir un algorithme de complexité raisonnable pour pouvoir construire un réseau expansif (linéaire ou non) dont l'alphabet se rapprocherait de $\beta(G)$ à partir de n'importe quel digraphe fortement connexe et couvrable.

Il serait aussi intéressant d'expliciter ce que la quantité $\beta(G)$ nous apprends sur la structure de G. Par exemple, le graphe que l'on a construit pour obtenir le plus grand $\beta(G)$ possible a une structure d'étoile. Il s'agit d'un graphe très régulier, dont un sommet a un degré entrant très important et par lequel « beaucoup d'informations doivent passer ». On peut se demander si l'un de ces critère est important pour avoir un grand paramètre β.

Enfin, dans ce chapitre, on a regardé plusieurs variantes de l'expansivité où on fait varier le type d'observations considérées. Toutefois, dans toutes ces variantes, il fallait qu'à partir d'un certain nombre d'observations, on puisse déduire la totalité de la configuration initiale du réseau d'automates. On pourrait être moins exigent, et ne pas demander à connaitre toute la configuration initiale, mais plutôt le cycle limite que le réseau a atteint.

Deuxième partie

Calcul séquentiel

Chapitre 5

Coût de séquentialisation

5.1 Introduction

Il y a eu beaucoup de travaux étudiant les différentes valeurs de f^W, avec f un réseau d'automates, selon la nature du mode de mise à jour W [6, 7, 10, 11, 12, 74]. Dans ce chapitre, comme dans [152], on se pose le problème inverse. On considère un réseau $h \in \mathrm{F}(n, q)$ et on cherche un réseau $f \in \mathrm{F}(n, q)$ ainsi qu'un mode de mise à jour séquentiel $w \in \mathbb{S}_{[n]}$ tels que $h = f^w$. Toutefois, il n'est pas toujours possible de trouver un tel f. En effet, considérons la fonction $h \in \mathrm{F}(2)$ définie par $h : x_1x_2 \mapsto x_2x_1$. Considérons par l'absurde qu'il existe $f \in \mathrm{F}(2)$ et $w \in \mathbb{S}_{[n]}$ tel que $f^w = h$. On peut considérer sans perte de généralité que $w = (1, 2)$. De ce fait,

$$x_2x_1 = h(x) = f^{(1,2)}(x) = f^{(2)}(f_1(x)x_2) = f_1(x)f_2(f_1(x)x_2).$$

Par conséquent, $f_1(x) = x_2$ et donc $x_2x_1 = x_2f_2(x_2x_2)$. Il en résulte que 01 et 11 sont tous les deux égaux à $1f_2(11)$ ce qui est une contradiction. Toutefois, on peut considérer un réseau $f \in \mathrm{F}(3)$ défini par $f : x \mapsto x_2x_3x_1$ et le mode de mise à jour séquentiel $w := (3, 1, 2) \in \mathbb{S}_{[3]}$. On a alors,

$$x_1x_2x_3 \xrightarrow{f^{(3)}} x_1x_2x_1 \xrightarrow{f^{(1)}} x_2x_2x_1 \xrightarrow{f^{(2)}} x_2x_1x_1.$$

Certes, la taille de f dépasse celle de h et on n'a donc clairement pas $f^w = h$. Mais, à défaut d'être égaux, on voit que f^w et h ont la même dynamique si on se concentre sur les deux premières coordonnées. On dit alors que f *simule* h grâce à w. On dit aussi que f *séquentialise* h avec une *mémoire additionnelle* de 1.

Plus formellement et plus généralement, étant donné un entier $m \geq n$, on dit que $f \in \mathrm{F}(m, q)$ *séquentialise* $h \in \mathrm{F}(n, q)$ avec une *mémoire additionnelle* de $m - n$ si

$$\mathrm{pr}_{[n]} \circ f^w = h \circ \mathrm{pr}_{[n]} \text{ pour un certain } w \in \mathbb{S}_{[m]}.$$

Ce chapitre porte sur le *coût de séquentialisation* $\boldsymbol{\kappa}(h)$ d'un réseau h, c'est-à-dire la quantité de mémoire additionnelle du plus petit réseau f qui séquentialise h. Comme on vient de le voir, il existe des réseaux h tel que $\boldsymbol{\kappa}(h) > 0$. On peut également se convaincre que $\boldsymbol{\kappa}(h) \leq n$ pour tout réseau $h \in \mathrm{F}(n, q)$. En effet, à chaque réseau $h \in \mathrm{F}(n, q)$ on peut associer un réseau $f \in \mathrm{F}(2n, q)$ tel que

$$f : xy = x_1 \dots x_n y_1 \dots y_n \mapsto h(y)x$$

" et une permutation $w = (n + 1, n + 2, \dots, 2n, 1, 2, \dots, n) \in \mathbb{S}_{[2n]}$. On peut voir que $\mathrm{pr}_{[n]} \circ f^w = h \circ \mathrm{pr}_{[n]}$ et donc que f séquentialise h avec une mémoire additionnelle de n.

Le premier résultat présenté dans ce chapitre montre qu'il est possible de séquentialiser tout réseau avec une mémoire additionelle bien inférieure.

Théorème 20. *Pour tout réseau $h \in \mathrm{F}(n,q)$, on a $\boldsymbol{\kappa}(h) \leq \lceil n/2 + \log_q(n/2+1) \rceil$.*

Le deuxième résultat important est que lorsque q est supérieur à 4 et n tend vers l'infini, la borne supérieure est en fait optimale.

Théorème 21. *Si $q \geq 4$, alors il existe $h \in \mathrm{F}(n,q)$ tel que $\boldsymbol{\kappa}(h) \geq \lfloor n/2 - \log_q(n) \rfloor - \mathcal{O}(1)$. Si $q \geq 2$ alors il existe $h \in \mathrm{F}(n,q)$ tel que $\boldsymbol{\kappa}(h) \geq \lfloor n/3 \rfloor$. Enfin, si $q = 3$ et n tend vers l'infini alors il existe $h \in \mathrm{F}(n,q)$ tel que $\boldsymbol{\kappa}(h) \geq \frac{n}{1+2\log_3(2)} - \mathcal{O}(\sqrt{n}) \geq \frac{n}{2{,}26} - \mathcal{O}(\sqrt{n})$.*

Il existe une autre manière de séquentialiser le réseau $h \in \mathrm{F}(n)$ tel que $h : x_1x_2 \mapsto x_2x_1$ sans mémoire additionnelle. Par exemple, on peut considérer le réseau $f \in \mathrm{F}(n)$ tel que $f : x \mapsto (x_1 \oplus x_2, x_1 \oplus x_2)$ et f^w simule alors h. En effet,

$$x_1x_2 \xrightarrow{f^{(1)}} (x_1 \oplus x_2, x_2) \xrightarrow{f^{(2)}} (x_1 \oplus x_2, x_1) \xrightarrow{f^{(1)}} (x_2, x_1)$$

Ceci ne signifie pas que le coût de séquentialisation de h est 0, car le mot $w = (1,2,1)$ ne décrit pas un mode de mise à jour séquentiel. En effet, la coordonnée 1 est mise à jour deux fois. Une telle répétition permet une réduction drastique de la mémoire additionnelle nécessaire pour simuler h. En fait, comme on le verra dans le prochain chapitre, pour tout n et q, on peut construire un réseau $f \in \mathrm{F}(n+1,q)$ complet pour $\mathrm{F}(n,q)$. En d'autre termes, à chaque réseau $h \in \mathrm{F}(n,q)$ on peut associer un mot $w \in [n+1]^*$ tel que $\mathrm{pr}_{[n]} \circ f^w = h \circ \mathrm{pr}_{[n]}$. En revanche, on verra que dans ce cas-là, le mot w utilisé peut être très grand.

À l'inverse, on va montrer qu'en s'autorisant plus de mémoire additionnelle, il est possible de simuler h avec un mot bien plus court. En fait, la taille minimale d'un tel mot est intimement reliée au coût de séquentialisation de h. Plus précisément, on considère un réseau $h \in \mathrm{F}(n,q)$ et l'on note $\mathcal{L}^*(h)$ le plus petit entier ℓ tel qu'il existe $m \geq n$, un mot w sur l'alphabet $[m]$ de taille ℓ et un réseau $f \in \mathrm{F}(m,q)$ tels que f simule h grâce à w. On va montrer que ce nombre $\mathcal{L}^*(h)$ est équivalent à la *complexité procédurale de h avec mémoire illimitée*, définie ici comme dans [68]. Notre troisième résultat concerne le lien entre $\boldsymbol{\kappa}(h)$ et $\mathcal{L}^*(h)$ et montre qu'en général, ces deux valeurs sont égales.

Théorème 22. *Pour tout $h \in \mathrm{F}(n,q)$, on a $\mathcal{L}^*(h) = \Omega(h) + \boldsymbol{\kappa}(h)$ avec $\Omega(h)$ le nombre de coordonnées $i \in [n]$ telles que $h^{(i)}$ n'est pas la fonction identité.*

Le dernier résultat de ce chapitre porte sur les graphes d'interaction. On montre que l'on peut borner supérieurement le coût de séquentialisation d'un réseau grâce à la *directed vertex separation* (DVS) de son graphe d'interaction. On pourrait traduire *directed vertex separation* par *séparation dirigée en sommet*.

Théorème 23. *Pour tout graphe d'interaction G, pour tout $h \in \mathrm{F}(G,*)$, $\boldsymbol{\kappa}(h)$ est inférieur à la DVS de G.*

Ce chapitre est organisé comme suit. Dans la section 5.2, on donne les notations et définitions spécifiques au chapitre. Dans la section 5.3, on montre que le problème de trouver le coût de séquentialisation d'un réseau est relié à celui de trouver le nombre chromatique d'un graphe nommé le *graphe de confusion*. Dans la section 5.4, on étudie une version raffinée du problème et on prouve la borne supérieure de $\boldsymbol{\kappa}(h)$ donnée dans théorème 20. Dans la section 5.5, on prouve cette fois les bornes inférieures de $\boldsymbol{\kappa}(h)$ données dans le théorème 21. Dans la section 5.6 on montre le lien entre le coût de séquentialisation et la complexité procédurale. Le théorème 22 est prouvé ici. La section 5.7 parle de borner le coût de séquentialisation d'un réseau h à partir de son graphe d'interaction. C'est ici que l'on prouve le théorème 23. Enfin, la section 5.8 offre une conclusion et quelques pistes de recherches.

Le travail présenté dans ce chapitre a fait l'objet d'une première publication en conférence [27]. Cette première publication ne porte que sur le cas booléen, les résultats donnés y sont moins forts et les problèmes abordés dans les sections 5.5, 5.6 et 22 n'y figurent pas. Un papier avec tous ces nouveaux résultats est en cours d'écriture.

5.2 Définitions et notations

Premièrement, rappelons les notions de simulation et de séquentialisation abordées plus tôt.

Définition 6 (Simulation et séquentialisation)**.** *Soit* $h \in \mathrm{F}(n,q)$*,* $f \in \mathrm{F}(m,q)$ *avec* $m \geq n$*. On dit que* f simule *ou* séquentialise h *grâce à* $w \in \mathbb{S}_{[m]}$*, si* $\mathrm{pr}_{[n]} \circ f^w = h \circ \mathrm{pr}_{[n]}$*.*

Remarque 4. *Tous les résultats de ce chapitre resteraient vrais si la définition de simulation qui exige que* $\mathrm{pr}_{[n]} \circ f^w = h \circ pr_{[n]}$ *était remplacée par la définition plus générale suivante. Il existe* $I \subseteq [m]$*, avec* $|I| = n$ *tel que* $\mathrm{pr}_I \circ f^w = h \circ \mathrm{pr}_I$*.*

Ensuite, on va rappeler la définition du coût de séquentialisation.

Définition 7 (Coût de séquentialisation)**.** *Soit* $h \in \mathrm{F}(n,q)$*. Le* coût de séquentialisation *de* h *(noté* $\boldsymbol{\kappa}(h)$*) est le plus petit entier* k *tel qu'il existe un réseau* $f \in \mathrm{F}(n+k,q)$ *qui séquentialise* h*.*

De plus on note,

$$\boldsymbol{\kappa}_{n,q} := \max\{\boldsymbol{\kappa}(h) \mid h \in \mathrm{F}(n,q)\}.$$

Pour étudier le coût de séquentialisation de h, il est pratique d'étudier une version plus raffinée de ce coût où on ajoute quelques contraintes sur le mode de mise à jour séquentiel.

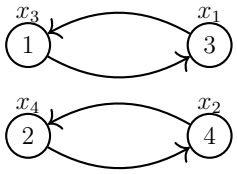


FIGURE 5.1 – Réseau h de l'exemple 2.

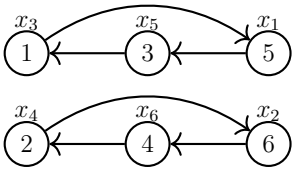


FIGURE 5.2 – Réseau f de l'exemple 2.

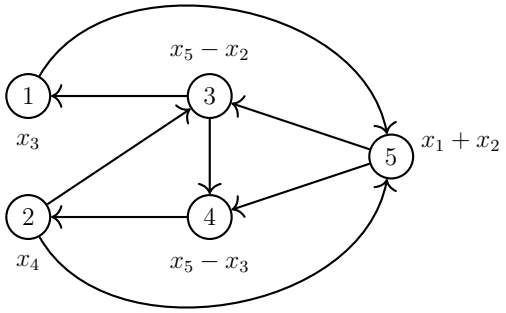


FIGURE 5.3 – Réseau g de l'exemple 2

Définition 8 (Coût de séquentialisation dans l'ordre u)**.** *Soit $h \in \mathrm{F}(n,q)$ et $u \in \mathbb{S}_{[n]}$. Le* coût de séquentialisation de h dans l'ordre u *(noté $\boldsymbol{\kappa}(h,u)$) est le plus petit entier k tel qu'il existe $f \in \mathrm{F}(n+k,q)$ et $w \in \mathbb{S}_{[n+k]}$ tel que f simule h grâce à w et que u soit un sous-mot de w.*

On écrit

$$\tilde{\boldsymbol{\kappa}}_{n,q} := \max\{\boldsymbol{\kappa}(h,u) \mid h \in \mathrm{F}(n,q) \text{ et } u \in \mathbb{S}_{[n]}\}.$$

Clairement, $\boldsymbol{\kappa}(h) = \min\{\boldsymbol{\kappa}(h,u) \mid u \in \mathbb{S}_{[n]}\}$ et donc $\boldsymbol{\kappa}_{n,q} \leq \tilde{\boldsymbol{\kappa}}_{n,q}$. L'exemple 2 ci-dessous montre que la différence entre $\boldsymbol{\kappa}(h)$ et $\boldsymbol{\kappa}(h,u)$ peut être grande.

Exemple 2. Considérons le réseau $h \in \mathrm{F}(n,q)$ représenté dans la figure 5.1. Le réseau h est de taille $n = 4$ et il permute 2 paires de valeurs. On a spécifiquement,

$$h : x \mapsto x_3x_4x_1x_2 = x_{[3,4]}x_{[1,2]}.$$

Maintenant, considérons le mode de mise à jour séquentiel canonique $u = (1,2,\ldots,6)$. Nous voudrions trouver un réseau $f \in \mathrm{F}(6,q)$ et un mot $w \in \mathbb{S}_{[6]}$ tel que f séquentialise h grâce à w dans l'ordre u. Nous allons montrer que cela peut se faire avec une mémoire additionnelle de 2. Prenons $f \in \mathrm{F}(6,q)$ qui réalise une permutation circulaire comme illustrée sur la figure 5.2. Autrement dit, on a $f : x \mapsto x_{[3,4]}x_{[5,6]}x_{[1,2]}$. Considérons la permutation $w = (5,6,1,2,3,4) \in \mathbb{S}_{[6]}$ qui a comme préfixe une permutation des $n/2$ coordonnées additionnelles de f et qui a u comme suffixe. Maintenant, on peut voir qu'une application de f^w nous donne

$$x = x_{[1,2]}x_{[3,4]}x_{[5,6]} \xrightarrow{f^{(5,6)}} x_{[1,2]}x_{[3,4]}x_{[1,2]} \xrightarrow{f^{(1,2)}} x_{[3,4]}x_{[3,4]}x_{[1,2]} \xrightarrow{f^{(3,4)}} x_{[3,4]}x_{[1,2]}x_{[1,2]}.$$

Nous avons donc $\mathrm{pr}_{[n]} \circ f^w = h \circ \mathrm{pr}_{[n]}$. Donc, f qui est de taille $n+2$ séquentialise h dans l'ordre u et ceci montre que $\boldsymbol{\kappa}(h,u) \leq 2$. De plus, on verra plus tard dans ce chapitre (dans la proposition 3 de la section 5.3 spécifiquement) qu'il n'y a pas de plus petit réseau f qui convienne. Par conséquent, on a $\boldsymbol{\kappa}(h,u) = n/2 = 2$.

Ensuite, on peut montrer qu'il existe $g \in \mathrm{F}(5,q)$ (qui a donc une mémoire additionnelle de taille 1) et $v \in \mathbb{S}_{[5]}$ tel que g simule h grâce à v (mais sans respecter l'ordre u). Premièrement, prenons $v = (5,1,3,2,4)$ qui, au lieu de mettre à jour les coordonnées de $[n]$ dans l'ordre u, met à jour les paires de coordonnées $(1,3)$ et $(2,4)$ l'une après l'autre. Alors, on définit le réseau $g \in \mathrm{F}(7,q)$ comme

$$g : x \mapsto (x_3, x_4, x_5 - x_2, x_5 - x_3, x_1 + x_2).$$

Comme toujours, les additions et soustractions sont faites modulo q. Le réseau g est représenté dans la figure 5.3. On peut vérifier que $g^v : x \mapsto x_{[3,4]}x_{[1,2]}(x_1 + x_2)$. Donc, $\mathrm{pr}_{[n]} \circ g^v = h \circ \mathrm{pr}_{[n]}$. Par conséquent, g séquentialise h et donc $\boldsymbol{\kappa}(h) \leq 1$.

De manière plus générale, on peut montrer que, pour tout entier n pair et $q \geq 2$, on a $\boldsymbol{\kappa}(h,(1,2,\ldots,n)) = n/2$ alors que $\boldsymbol{\kappa}(h) \leq 1$ avec $h \in \mathrm{F}(n,q)$ le réseau défini par $h : x \mapsto x_{[\frac{n}{2}+1,n]}x_{[\frac{n}{2}]}$.

5.3 Graphe de confusion

Dans cette section, on étudie une manière de traduire le problème de trouver le coût de séquentialisation d'un réseau h en un problème plus simple de coloration de graphe. Pour faire ceci, on va définir un graphe : le *graphe de confusion*. Considérons la fonction $h \in \mathrm{F}(2)$ tel que $h : x_1x_2 \mapsto x_2x_1$ et la permutation $u = (1,2) \in \mathbb{S}_{[2]}$. Comme on l'a vu dans l'introduction, il n'y a pas de fonction f tel que $f^u = h$. C'est parce que $f \in \mathrm{F}(2)$ ne peut pas avoir en même temps les deux propriétés ci-dessous.

— $f^u(\mathtt{01}) = h(\mathtt{01})$ $(= \mathtt{10})$ et,

— $f^u(\mathtt{11}) = h(\mathtt{11})$ $(= \mathtt{11})$.

Par conséquent, on pourrait dire que le couple de configurations $\{\mathtt{01}, \mathtt{11}\}$ est un obstacle à la séquentialisation sans mémoire de h dans l'ordre u. Le graphe de confusion $G_{h,u}$ définit ci-dessous va représenter chaque telle paire de configurations $\{x,y\}$ à l'aide d'une arête $\{x,y\}$. On peut voir que si $G_{h,u}$ a la moindre arête, on a $\boldsymbol{\kappa}(h,u) > 0$, mais le lemme 21 va plus loin et montre que l'on peut calculer $\boldsymbol{\kappa}(h,u)$ directement depuis $G_{h,u}$.

Définition 9 (Graphe de confusion)**.** *Considérons $h \in \mathrm{F}(n,q)$ et la permutation $u \in \mathbb{S}_{[n]}$. Le* graphe de confusion *$G_{h,u}$ est le graphe dont les sommets sont les configurations de $\mathbb{A}^n$ et tel que deux configurations $x, x' \in \mathbb{A}^n$ sont voisines dans $G_{h,u}$ si et seulement si*

— *$h(x) \neq h(x')$, et*

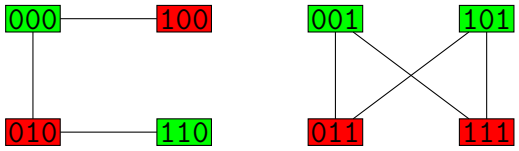
— $\exists\ i \in [n],\ h^{\{u_1,\dots,u_i\}}(x) = h^{\{u_1,\dots,u_i\}}(x')$.

Un exemple de graphe de confusion est donné dans la figure 5.4. Dans cet exemple, les configurations 000 et 100 sont voisines dans $G_{h,u}$ car $h^{\{1,2\}}(000) = 000 = h^{\{1,2\}}(100)$ et $h(000) = 000 \neq 001 = h(100)$.

À l'inverse, même si $h^{\{1,2\}}(100) = 000 = h^{\{1,2\}}(010)$, les configurations 100 et 010 ne sont pas voisines dans $G_{h,u}$ car $h(100) = 001 = h(010)$.

x	$h^{\{1\}}(x)$	$h^{\{1,2\}}(x)$	$h(x)$
000	**0**00	**00**0	**000**
100	**0**00	**00**0	**001**
010	**0**10	**00**0	**001**
110	**0**10	**01**0	**010**
001	**1**01	**11**1	**110**
101	**1**01	**11**1	**110**
011	**1**11	**11**1	**111**
111	**1**11	**11**1	**111**

(a) définition de h



(b) Graphe de confusion $G_{h,u}$

FIGURE 5.4 – Exemple de réseau $h \in \mathrm{F}(3)$ et son graphe de confusion pour l'ordre $u = (1, 2, 3)$.

Une coloration valide (des sommets) d'un graphe $G = (V, E)$ est une fonction $c : V \mapsto S$ (avec S un ensemble quelconque considéré comme un ensemble de couleur) tel que, pour tout $\{i, j\} \in E$, on a $c(i) \neq c(j)$. Par la suite, on note $\chi(G)$ le *nombre chromatique* d'un graphe G, c'est-à-dire la cardinalité minimum de l'ensemble S de couleurs d'une coloration valide (des sommets) de G. On note également $\omega(G)$ la taille de *la plus grosse clique* d'un graphe $G = (V, E)$, c'est-à-dire la cardinalité du plus gros ensemble $S \subseteq V$ tel que, pour tout $i, j \in S$, $\{i, j\} \in E$.

Lemme 21. *Pour tout $h \in \mathrm{F}(n, q)$ et $u \in \mathbb{S}_{[n]}$, on a $\boldsymbol{\kappa}(h, u) = \lceil \log_q(\chi(G_{h,u})) \rceil$.*

Démonstration. Posons $G_{h,u} = (\mathbb{A}^n, E)$ pour toute cette démonstration.

Borne inférieure : $\boldsymbol{\kappa}(h, u) \geq \lceil \log_q(\chi(G_{h,u})) \rceil$.

Sans perte de généralité, supposons que u est le mode de mise à jour séquentiel canonique $(1, 2, \dots, n)$. Posons $k = \boldsymbol{\kappa}(h, u)$, $m = n + k$ et $M = [n + 1, m]$. Considérons un réseau $f \in \mathrm{F}(m, q)$ et un mot $w \in \mathbb{S}_{[m]}$ tels que f séquentialise h dans l'ordre u grâce à w.

Posons $y = (0)^k$. Soit $c : \mathbb{A}^n \to \mathbb{A}^k$, la fonction de coloration du graphe $G_{h,u}$ telle que $c : x \mapsto f_M^w(xy)$. La fonction c utilise moins de q^k couleurs. Par conséquent, si c est une

coloration valide, on a $q^{\boldsymbol{\kappa}(h,u)} = q^k \geq \chi(G_{h,u})$ et donc $\boldsymbol{\kappa}(h,u) \geq \lceil \log_q(\chi(G_{h,u})) \rceil$. Il est donc suffisant de prouver que c est une coloration valide. En d'autres termes, pour tout $\{x, x'\} \in E$, on doit avoir $c(x) \neq c(x')$.

Considérons donc une arête $\{x, x'\} \in E$ et les configurations $z = xy$ et $z' = x'y$. Par l'absurde, supposons que $c(x) = c(x')$ et donc que $f_M^w(z) = f_M^w(z')$. Par définition du graphe de confusion $G_{h,u}$, on sait que $h(x) \neq h(x')$ et que $h^{\{u_1,\dots u_i\}}(x) = h^{\{u_1,\dots,u_i\}}(x')$ pour un certain $i \in [n]$. Considérons le plus grand i qui respecte cette condition. Remarquons que $i < n$ car $h(x) \neq h(x')$ et notons $I = \{u_1, \dots, u_i\}$. On a $h^{I \cup \{u_{i+1}\}}(x) \neq h^{I \cup \{u_{i+1}\}}(x')$ et donc $h_{u_{i+1}}(x) \neq h_{u_{i+1}}(x')$. Décomposons w en $v(u_{i+1})v'$. Prouvons que $f^v(z) = f^v(z')$. Premièrement, comme f simule h grâce à w, on a

$$f^v_{[n]}(z) = h^I(x) = h^I(x') = f^v_{[n]}(z').$$

De plus, pour tout coordonnée $j \in M$ mise à jour dans v' (et donc pas dans v), on a

$$f^v_j(z) = z_j = y_{j-n} = z'_j = f^v_j(z').$$

Enfin, pour toute coordonnée $j \in M$ apparaissant dans v on a $f^v_j(z) = c_{j-n}(x) = c_{j-n}(x') = f^v_j(z')$. Il découle des 3 égalités précédentes que $f^v(z) = f^v(z')$. Mais

$$f_{u_{i+1}} \circ f^v_{u_{i+1}}(z) = f^w_{u_{i+1}}(z) = h_{u_{i+1}}(x) \neq h_{u_{i+1}}(x') = f^w_{u_{i+1}}(z') = f_{u_{i+1}} \circ f^v(z').$$

Ceci pose une contradiction. Par conséquent, on a $f_M^w(z) \neq f_M^w(z')$. En conséquence, c est bien une coloration valide.

Borne supérieure : $\boldsymbol{\kappa}(h,u) \leq \lceil \log_q(\chi(G_{h,u})) \rceil$.

Soit $k = \lceil \log_q(\chi(G_{h,u})) \rceil$, $m = n + k$ et $M = [n+1, m]$. Soit $w \in \mathbb{S}_{[m]}$ tel que $w = (n+1, \dots, n+k)u$.

Considérons $c : \mathbb{A}^n \to \mathbb{A}^k$ une coloration valide de $G_{h,u}$. Pour tout $i \in [n]$ et pour toute configuration $z \in \mathbb{A}^m$, définissons l'ensemble $X_i(z) \subseteq \mathbb{A}^n$ comme suit. Premièrement, $X_1(z)$ est le singleton $\{z_{[n]}\}$ et ensuite $\forall i \in [2, n]$, $X_i(z) = \{x \in \mathbb{A}^n \mid h^{\{u_1 \dots u_{i-1}\}}(x) = z_{[n]}$ et $c(x) = z_{[n+1,m]}\}$.

Soit $f \in \mathrm{F}(m, q)$ tel que

— $f_M = c \circ \mathrm{pr}_{[n]}$, et

— $\forall i \in [n],\ f_{u_i} : z \mapsto \begin{cases} z_{u_i} & \text{si } X_i(z) = \emptyset \text{ et,} \\ h_{u_i}(x) \text{ avec } x \in X_i(z) & \text{sinon} \end{cases}$.

Prouvons que $\mathrm{pr}_{[n]} \circ f^w = h \circ \mathrm{pr}_{[n]}$. Soit $x \in \mathbb{A}^n$ et $z \in \mathbb{A}^m$ avec $z_{[n]} = x$. Par récurrence, montrons que

$$\forall i \in [0, n],\ f^{(w_1 \dots w_{k+i})}(z) = h^{\{u_1,\dots,u_i\}}(x)c(x).$$

Premièrement, pour $i = 0$, on a bien,

$$f^{(w_1 \dots w_k)}(z) = f^{(n+1,\dots,n+k)}(z) = xc(x).$$

Ensuite, prenons $i \in [n]$, posons $v = (w_1, \dots, w_{k+(i-1)})$ et supposons que,

$$f^v(z) = h^{\{u_1,\dots,u_{i-1}\}}(x)c(x).$$

On a donc $f_{w_{k+i}} \circ f^v(z) = h_{u_i}(x')$ avec $x' \in X_i(f^v(z))$. Remarquons que $x \in X_i(f^v(z))$ également.

Prouvons que $h_{u_i}(x') = h_{u_i}(x)$. Par l'absurde, supposons que $h_{u_i}(x') \neq h_{u_i}(x)$. Par conséquent, $h(x) \neq h(x')$. Toutefois, $x, x' \in X_i(f^v(z))$ et donc $h^{\{u_1,\dots,u_{i-1}\}}(x) = h^{\{u_1,\dots,u_{i-1}\}}(x')$ et $c(x) = c(x')$. Cela signifie que $\{x, x'\} \in E$, alors que x et x' ont la même couleur. C'est absurde. On a donc bien, $h_{u_i}(x') = h_{u_i}(x)$, $f^{(w_1 \dots w_{k+i})}(z) = h^{\{u_1,\dots,u_i\}}(x)$ et l'hypothèse de récurrence est prouvée. Par conséquent, $f^w(z) = h(x)c(x)$ et $\mathrm{pr}_{[n]} \circ f^w = h \circ \mathrm{pr}_{[n]}$. Comme f a k coordonnées additionnelles, on a $\boldsymbol{\kappa}(h, u) \leq k \leq \lceil \log_q(\chi(G_{h,u})) \rceil$.

En conclusion, $\boldsymbol{\kappa}(h, u) = \lceil \log_q(\chi(G_{h,u})) \rceil$.

□

Remarque 5. *En observant notre construction pour la borne supérieure, on remarque que le coût de séquentialisation minimum peut être atteint grâce à un mode de mise à jour séquentiel w qui met d'abord à jour toutes les coordonnées additionnelles, puis les coordonnées principales.*

Si on observe de nouveau l'exemple donné dans la figure 5.4, on peut voir que le nombre chromatique de $G_{h,u}$ est de 2. En effet, on peut observer qu'il existe une coloration valide de $G_{h,u}$ utilisant seulement 2 couleurs et, comme il y a au moins une arête dans $G_{h,u}$, il n'y a pas de coloration valide monochromatique. De ce fait, d'après le lemme 21, que l'on vient de prouver,

$$\boldsymbol{\kappa}(h, u) = \lceil \log_2(\chi(G_{h,u})) \rceil = \lceil \log_2(2) \rceil = 1.$$

La proposition 2 ci-dessous affirme que le coût de séquentialisation qui doit respecter un certain ordre ou non ($\tilde{\boldsymbol{\kappa}}_{n,q}$ ou $\boldsymbol{\kappa}_{n,q}$) est croissant en fonction de n. La preuve est très simple. Depuis un réseau $h \in \mathrm{F}(n, q)$ et une taille $n' > n$, il est facile de construire un réseau $h' \in \mathrm{F}(n', q)$ tel que $\boldsymbol{\kappa}(h) \leq \boldsymbol{\kappa}(h')$ (et également tel que, pour tout $u \in \mathbb{S}_{[n]}$ il existe $u' \in \mathbb{S}_{[n]}$ tel que $\boldsymbol{\kappa}(h, u) \leq \boldsymbol{\kappa}(h', u')$). Cette proposition sera entre autres utilisée pour simplifier la preuve que, pour tout n, $\boldsymbol{\kappa}_{n,q} \geq \lfloor n/4 \rfloor$. En effet, grâce à la proposition 2, il est suffisant de prouver que, pour tout k, $\boldsymbol{\kappa}_{4k,q} \geq k$.

Proposition 2. *Pour tout $n < n'$, on a $\tilde{\boldsymbol{\kappa}}_{n,q} \leq \tilde{\boldsymbol{\kappa}}_{n',q}$ et $\boldsymbol{\kappa}_{n,q} \leq \boldsymbol{\kappa}_{n',q}$.*

Démonstration. Soit $p = n' - n$, $I = [n+1, n']$ et $h \in \mathrm{F}(n,q)$. Soit $h' \in \mathrm{F}(n',q)$ tel que, pour tout $x' \in \mathbb{A}^{n'}$,

$$h'_{[n]}(x') = h(x'_{[n]}) \text{ et } h'_I(x') = (0)^p.$$

Considérons $u' \in \mathbb{S}_{[n']}$ et soit $u \in \mathbb{S}_{[n]}$ tel que u soit un sous-mot de u'. Prouvons que $G_{h',u'} = (\mathbb{A}^{n'}, E')$ restreint aux configurations de $\{x' \in \mathbb{A}^{n'} \mid x'_I = (0)^p\}$ est isomorphe au graphe de confusion $G_{h,u} = (\mathbb{A}^n, E)$. Considérons $x, y \in \mathbb{A}^n$ et $x', y' \in \mathbb{A}^{n'}$ tels que $x'_{[n]} = x$, $y'_{[n]} = y$ et $x'_I = y'_I = (0)^p$. On a alors, $h(x) \neq h(y)$ si et seulement si $h'(x') \neq h'(y')$. De plus, si $\{x, y\} \in E$ dans $G_{h,u}$ alors il existe $t \in [n]$ tel que $h^{(u_1,\dots,u_t)}(x) = h^{(u_1,\dots,u_t)}(y)$. On peut montrer que l'on a alors $\{x', y'\} \in E'$. En effet, pour t' tel que $u_{t'} = u_t$, on a

$$h'^{(u'_1,\dots,u'_{t'})}(x') = h^{(u_1,\dots,u_t)}(x)(0)^p = h^{(u_1,\dots,u_t)}(y)(0)^p = h'^{(u'_1,\dots,u'_{t'})}(y').$$

On peut montrer que $\{x', y'\} \in E' \Rightarrow \{x, y\} \in E$ de manière totalement symétrique. De ce fait, $G_{h',u'}$ contient un sous-graphe isomorphe à $G_{h,u}$. Clairement, $\chi(G_{h',u'}) \geq \chi(G_{h,u})$. Par conséquent, pour tout $u \in \mathbb{S}_{[n]}$, $\boldsymbol{\kappa}(h,u) \leq \boldsymbol{\kappa}(h',u')$ pour un certain $u' \in \mathbb{S}_{[n]}$. En particulier, prenons $h \in \mathrm{F}(n,q)$ et $u \in \mathbb{S}_{[n]}$ tels que $\tilde{\boldsymbol{\kappa}}_{n,q} = \tilde{\boldsymbol{\kappa}}(h,u)$. On a clairement $\tilde{\boldsymbol{\kappa}}_{n,q} = \tilde{\boldsymbol{\kappa}}(h,u) \leq \tilde{\boldsymbol{\kappa}}(h',u') \leq \tilde{\boldsymbol{\kappa}}_{n',q}$. De la même façon, en prenant $h \in \mathrm{F}(n,q)$ tel que $\boldsymbol{\kappa}_{n,q} = \boldsymbol{\kappa}(h)$, on a $\boldsymbol{\kappa}_{n,q} = \boldsymbol{\kappa}(h) \leq \boldsymbol{\kappa}(h') = \boldsymbol{\kappa}_{n',q}$. □

5.4 Coût de séquentialisation dans un certain ordre

Dans cette section, on étudie le coût de séquentialisation d'un réseau h dans un certain ordre u. On montre d'abord dans le théorème 20 que l'on peut trouver une borne supérieure à $\tilde{\boldsymbol{\kappa}}_{n,q}$. Clairement elle est également une borne supérieure pour $\boldsymbol{\kappa}_{n,q}$. C'est en fait la meilleure que l'on connaisse actuellement. Deuxièmement, on donne dans la proposition 3 une borne inférieure pour $\tilde{\boldsymbol{\kappa}}_{n,q}$ très proche de sa borne supérieure.

Pour prouver le théorème 20, on regroupe toutes les configurations d'un graphe $G_{h,u}$ qui sont égales sur leurs secondes moitiés ($x_{\{u_{n/2+1},\dots,u_n\}} = x'_{\{u_{n/2+1},\dots,u_n\}}$) et ont la même image ($h(x) = h(x')$) ensemble. On colorie toutes les configurations d'un même groupe de la même couleur. De plus, on montre que le degré maximum de ce graphe « factorisé » est inférieur à $\lceil (n/2+1)q^{n/2} \rceil$. On peut conclure en utilisant le théorème de Brooks qui affirme que le nombre chromatique d'un graphe est au plus son degré maximum plus un (voir le théorème 14.4 du livre [20]).

Théorème 20. $\boldsymbol{\kappa}_{n,q} \leq \tilde{\boldsymbol{\kappa}}_{n,q} \leq \lceil n/2 + \log_q(n/2+1) \rceil$.

Démonstration. Soit $h \in \mathrm{F}(n,q)$ et $u \in \mathbb{S}_{[n]}$. Prouvons que $\boldsymbol{\kappa}(h,u) \leq \lceil n/2 + \log_q(n/2+1) \rceil$. Sans perte de généralité, supposons que u est le mode de mise à jour séquentiel canonique

$(1, 2, \ldots, n)$. Posons $G_{h,u} = (\mathbb{A}, E)$. Soit $X = \{X_1, \ldots, X_p\}$ une partition de $\mathbb{A}^n$, telle que x, x' sont dans le même ensemble X_i si et seulement si les deux conditions suivantes sont vraies.

— Les deux configurations sont égales sur la seconde partie des coordonnées mises à jour par u. En d'autres termes, $x_{\{u(n/2+1),\ldots,u(n)\}} = {x'}_{\{u(n/2+1),\ldots,u(n)\}}$ ou plus simplement, $x_{]n/2,n]} = {x'}_{]n/2,n]}$ car on a supposé que $u = (1, 2, \ldots, n)$.

— Elles ont la même image par h. En d'autres mots, $h(x) = h(x')$.

Pour tout $x \in \mathbb{A}^n$, écrivons $X(x)$ l'ensemble $X_i \in X$ qui contient x. Considérons $x^{(1)} \in X_1,\ x^{(2)} \in X_2,\ \ldots,\ x^{(p)} \in X_p$.

Soit $G' = (X, E')$, le graphe tel que les ensembles X_i et $X_{i'}$ sont voisins dans G' si et seulement s'il existe deux configurations $x \in X_i$ et $x' \in X_{i'}$ voisines dans $G_{h,u}$.

Prouvons que le degré de G' est d'au plus $(n/2 + 1)q^{n/2}$. Soit X_ℓ, le sommet de degré maximum dans G'. Considérons son ensemble de voisins $N \subseteq X$. Par définition, si $X_j \in N$ alors il existe deux configurations $x \in X_\ell, x' \in X_j$ telles que $\exists i \in [n], h^{[i]}(x) = h^{[i]}(x')$ et $h(x) \neq h(x')$.

Décomposons N en $n/2 + 1$ ensembles (pas forcément disjoints) comme suit.

— L'ensemble $N_{[n/2]} \subseteq X$ est l'ensemble des $X_j \in X$ tels qu'il existe un entier $i \in [n/2]$ et deux configurations $x \in X_\ell$ et $x' \in X_j$ tels que

$$h^{[i]}(x) = h^{[i]}(x') \text{ et } h(x) \neq h(x').$$

Comme $h^{[i]}(x') = h^{[i]}(x)$, on a ${x'}_{]i,n]} = x_{]i,n]}$ et ${x'}_{]n/2,n]} = x_{]n/2,n]} = x^{(\ell)}_{]n/2,n]}$ car $i \leq n/2$. En d'autres termes, pour tout ensemble $X_j \in N_{[n/2]}$, on a une configuration $x' \in X_j$ telle que ${x'}_{]n/2,n]} = x^{(\ell)}_{]n/2,n]}$. On remarque, qu'il n'y a que $q^{n/2}$ configurations x' avec cette caractéristique. Donc, $|N_{[n/2]}| \leq q^{n/2}$.

— Pour tout $i \in [n/2 + 1, n]$, soit $N_i \subseteq X$ tel que $X_j \in N_i$ si et seulement si

$$\exists x \in X_1, x' \in X_j \text{ telles que } h^{[i]}(x) = h^{[i]}(x') \text{ et } h(x) \neq h(x').$$

Soit $X_j \in N_i$ et soient les deux configurations $x \in X_\ell$ et $x' \in X_j$ telles que $h^{[i]}(x) = h^{[i]}(x')$. Donc, on a $x^{(\ell)}_{]i,n]} = {x'}_{]i,n]} = x_{]i,n]} = x^{(j)}_{]i,n]}$ car $i > n/2$. Mais, la valeur de $x^{(j)}_{[n/2+1,n]}$ est fixée sur l'intervalle $[i, n]$ et peut seulement varier sur l'intervalle $[n/2 + 1, i]$. Donc, la deuxième moitié de $x^{(j)}$ peut seulement prendre $q^{i-n/2}$ valeurs. De plus, on sait que $h(x^{(\ell)}) = h^{[i]}(x)$, $h^{[i]}(x) = h^{[i]}(x')$ et $h(x') = h(x^{(j)})$. Par conséquent, $h_{[i]}(x^{(\ell)}) = h_{[i]}(x) = h_{[i]}(x') = h_{[i]}(x^{(j)})$. Donc, la valeur $h(x^{(j)})$ est fixée sur l'intervalle $[i]$ et peut seulement varier sur l'intervalle $[i, n]$. De ce fait, $h(x^{(j)})$ peut seulement prendre q^{n-i} valeurs différentes. Maintenant, si deux configurations ont la même image par h et sont égales sur leur seconde moitié, alors elles appartiennent au même ensemble X_j. Donc, $|N_i| \leq q^{i-n/2} \cdot q^{n-i} = q^{n/2}$.

On a donc $N = N_{\lceil n/2 \rceil} \cup N_{n/2+1} \cup \cdots \cup N_n$. De ce fait, $|N| \leq (n/2+1)q^{n/2}$. Par conséquent, le degré de X^1 dans G' est au plus $(n/2+1)q^{n/2}$. En fait, le degré de X^1 est strictement inférieur à $(n/2+1)q^{n/2}$ car X^1 n'est pas voisin de lui-même et a pourtant été compté quand on a borné la taille de N. Par conséquent, $\chi(G') \leq \Delta(G')+1 \leq (n/2+1)q^{n/2}$ avec $\Delta(G')$ le degré maximum de G'. On peut voir que toute coloration du graphe G' nous donne une coloration valide du graphe de confusion $G_{h,u}$. En effet, on peut colorier chaque configuration $x \in X_i$ dans $G_{h,u}$ de la même façon que X_i dans G'. Si deux configurations x et x' sont adjacentes dans le graphe de confusion $G_{h,u}$, alors $X(x)$ et $X(x')$ sont adjacents dans G' et n'ont pas la même couleur. Donc, $\chi(G_{h,u}) \leq \chi(G') \leq (n/2+1) \cdot q^{n/2}$. Par conséquent, d'après le lemme 21, on a, $\boldsymbol{\kappa}(h,u) \leq \lceil n/2 + \log_q(n/2+1) \rceil$. De ce fait, $\tilde{\boldsymbol{\kappa}}_{n,q} \leq \lceil n/2 + \log_q(n/2+1) \rceil$. □

La proposition 3 ci-dessous donne une borne inférieure pour $\tilde{\boldsymbol{\kappa}}_{n,q}$. Pour la prouver, on construit un couple (h,u) tel que $G_{h,u}$ possède une clique de taille $q^{n/2}$. Il est bien connu que le nombre chromatique $\chi(G)$ d'un graphe G est au moins égal à la taille de sa plus grosse clique $\omega(G)$. Par conséquent, $\chi(G_{h,u}) \geq q^{n/2}$, $\boldsymbol{\kappa}(h,u) = \lceil \log(\chi(G_{h,u})) \rceil \geq n/2$, et on en déduit la proposition 3.

Proposition 3. $\tilde{\boldsymbol{\kappa}}_{n,q} \geq \lfloor n/2 \rfloor$.

Démonstration. Soit $k := \lfloor n/2 \rfloor$. Soit $h \in \mathrm{F}(n,q)$ le réseau tel que :

- $\forall i \in [k],\ h_i : x \mapsto x_{i+k}$,
- $\forall i \in [k+1, 2k],\ h_i : x \mapsto x_{i-k}$, et
- si n est impair $h_n : x \mapsto x_n$.

On prend le mode de mise à jour séquentiel canonique $u = (1, 2, \ldots, n)$. Soit $X := \{x \in \mathbb{A}^n \mid x_{[k+1,n]} = (\mathtt{0})^{n-k}\}$ l'ensemble des configurations qui sont composées uniquement de $\mathtt{0}$ sur leur deuxième moitié. Prouvons que X est une clique dans $G_{h,u} = (\mathbb{A}^n, E)$. Pour ça, considérons deux configurations distinctes $x, x' \in X$ et prouvons que $\{x, x'\} \in E$. On a $x_{[k+1,n]} = (\mathtt{0})^{n-k} = {x'}_{[k+1,n]}$. De plus, $x_{[k]} \neq {x'}_{[k]}$. Il existe donc une coordonnée $i \in [k]$ telle que $x_i \neq {x'}_i$ et $h_{i+k}(x) = x_i \neq {x'}_i = h_{i+k}(x')$. Par conséquent, $h(x) \neq h(x')$. Toutefois, quand on met à jour la première moitié de leurs coordonnées, les configurations x et x' se transforment toutes les deux en $(\mathtt{0})^n$. En effet, $\forall i \in [k], f_i(x) = x_{i+k} = \mathtt{0}$. Donc, on a $h^{[k]}(x) = (\mathtt{0})^n = h^{[k]}(x')$. Par conséquent, $\{x, x'\} \in E$ et X est bien une clique. De plus, X est une clique de taille q^k. Donc, $\chi(G_{h,u}) \geq \omega(G) \geq q^k$ et $\boldsymbol{\kappa}(h,u) \geq \lceil \log_q(\chi(G_{h,u})) \rceil \geq \lceil \log_q(q^k) \rceil = k = \lfloor n/2 \rfloor$. De ce fait, $\tilde{\boldsymbol{\kappa}}_{n,q} \geq \lfloor n/2 \rfloor$. □

On remarque que le réseau donné dans la proposition 3 est bijectif. Dans le théorème 5 de [68], il est montré que si $h \in \mathrm{F}(n,q)$ est bijectif, alors pour tout $u \in \mathbb{S}_{[n]}$ on a $\boldsymbol{\kappa}(h,u) \leq k$ quand $n = 2k$. Par conséquent, on sait déjà que lorsque h est un réseau bijectif de taille $2k$, alors on a exactement $\boldsymbol{\kappa}(h,u) = k$. Cela nous amène à penser que c'est également vrai pour les réseaux non bijectifs ce qui nous fait poser le conjecture suivante.

Conjecture 3. *Pour tout réseau $h \in \mathrm{F}(2k,q)$ et tout mot $u \in \mathbb{S}_{[n]}$, on a $\boldsymbol{\kappa}(h,u) \leq k$.*

5.5 Borne inférieure sur le coût de séquentialisation

Le but de cette section est de construire des réseaux avec le plus grand coût de séquentialisation possible pour trouver une borne inférieure à $\boldsymbol{\kappa}_{n,q}$. Cette borne inférieure dépend de la valeur de q et est donnée dans le théorème 21 ci-dessous.

Théorème 21. *Pour tout $q \geq 2$, on a*

$$\boldsymbol{\kappa}_{n,q} \geq \lfloor n/3 \rfloor.$$

Pour tout $q \geq 4$, on a

$$\boldsymbol{\kappa}_{n,q} \geq n/2 - \log_q(n) - \mathcal{O}(1).$$

Enfin, si $q = 3$ et que n est suffisamment grand, on a

$$\boldsymbol{\kappa}_{n,q} \geq \frac{n}{1 + 2\log_3(2)} - \mathcal{O}(\sqrt{n}) \geq \frac{n}{2,26} - \mathcal{O}(\sqrt{n}).$$

Le théorème 21 est une conséquence directe des lemmes 22, 23, 24 et 25.

On définit la notation suivante. Pour tout ensemble I et tout entier k, $\binom{I}{k} := \{J \subseteq I \mid |J| = k\}$ désigne l'ensemble des sous-ensembles de I de taille k.

Dans le lemme 22, on considère les entiers k et $n \geq 2k$ et on suppose que les hypothèses suivantes sont respectées. Premièrement, chaque ensemble $I_\ell \in \binom{[2k]}{k}$ est associé à un ensemble $X_\ell \subseteq \mathbb{A}^n$ de q^k configurations qui diffèrent seulement sur I_ℓ. Deuxièmement, les ensembles $X_1, \ldots, X_{\binom{2k}{k}}$ sont disjoints. Le lemme 22 affirme que dans ce cas, il existe un réseau $h \in \mathrm{F}(n,q)$ avec les deux propriétés suivantes. D'abord, h n'a pas de fonction locale triviale $h_i : x \mapsto x_i$. Ensuite, pour tout $u \in \mathbb{S}_{[n]}$, il existe un ensemble correspondant X_ℓ qui est une clique dans le graphe de confusion $G_{h,u}$. Par conséquent, $\boldsymbol{\kappa}(h,u) \geq k$ et $\boldsymbol{\kappa}(h) \geq k$.

Lemme 22. *Soient $n \geq 2k$, $s = \binom{2k}{k}$, $\binom{[2k]}{k} = \{I_1, \ldots, I_s\}$ et $a^1, \ldots, a^s \in \mathbb{A}^n$. Pour tout $\ell \in [s]$, soit*

$$X_\ell = \{x \in \mathbb{A}^n \mid x_{[n]\setminus I_\ell} = a^\ell_{[n]\setminus I_\ell}\}.$$

Si les ensembles $X_1, \ldots, X_s$ sont disjoints alors il existe un réseau $h \in \mathrm{F}(n,q)$ tel que $\Omega(h) = n$ et $\boldsymbol{\kappa}(h) \geq k$.

Démonstration. Soit $M = [2k+1, n]$. Pour tout $\ell \in [s]$, soit $I'_\ell = [2k] \setminus I_\ell$. Définissons $h \in \mathrm{F}(n,q)$ comme suit. Soit $x \in \mathbb{A}^n$. Si $x \in X_\ell$ pour un $\ell \in [s]$ alors $h_{[n]\setminus I'_\ell}(x) = (0)^{n-k}$ et $h_{I'_\ell}(x) = x_{I_\ell}$. Sinon, $h(x) = (0)^n$.

On peut vérifier que h n'a aucune fonction locale triviale. En effet, pour tout $i \in M$, la fonction locale h_i est non triviale car h_i retourne toujours 0. De plus, pour tout $i \in [k]$

(*resp.* $i \in [k+1, 2k]$), soit $\ell \in [s]$ tel que $I_\ell = [k+1, 2k]$ (*resp.* $I_\ell = [k]$) et soient deux configurations $x, y \in X^\ell$ telles que $x_{I_\ell} = (\mathtt{0})^k$ et $y_{I_\ell} = (\mathtt{1})^k$. On a $x_i = y_i = a_i^\ell$ mais $h_i(x) = \mathtt{0} \neq \mathtt{1} = h_i(y)$. Par conséquent, h_i est non triviale. Il en résulte que h n'a pas de fonction locale triviale et que $\Omega(h) = n$.

Maintenant, prouvons que $\boldsymbol{\kappa}(h) \geq k$. Pour ceci, il nous suffit de prouver que, pour tout $u \in \mathbb{S}_{[n]}$, on a $\boldsymbol{\kappa}(h, u) \geq k$. Considérons une permutation $u \in \mathbb{S}_{[n]}$. Soit $\ell \in [s]$ tel que I_ℓ est l'ensemble composé des k premières valeurs de u comprises dans $[2k]$. En d'autres termes, il existe t tel que $\{u_1, \ldots, u_t\} \cap [2k] = I_\ell$ et $\{u_{t+1}, \ldots, u_n\} \cap [2k] = I'_\ell$. Prouvons que X_ℓ est une clique dans le graphe de confusion $G_{h,u} = (\mathbb{A}^n, E)$. Considérons une paire de configurations $\{x, x'\} \subseteq X_\ell$. On peut prouver que $\{x, x'\} \in E$ comme suit.

— On a $h_{\{u_1,\ldots,u_t\}}(x) = (\mathtt{0})^t = h_{\{u_1,\ldots,u_t\}}(x')$ car

$$h_{[n]\setminus I'_\ell}(x) = (\mathtt{0})^{n-k} = h_{[n]\setminus I'_\ell}(x') \text{ et } \{u_1, \ldots, u_t\} \subseteq [n] \setminus I'_\ell.$$

— On a $x_{\{u_{t+1},\ldots,u_n\}} = a^\ell_{\{u_{t+1},\ldots,u_n\}} = {x'}_{\{u_{t+1},\ldots,u_n\}}$ car

$$x_{[n]\setminus I_\ell} = a^\ell_{[n]\setminus I_\ell} = {x'}_{[n]\setminus I_\ell}, \text{ et } \{u_{t+1}, \ldots, u_n\} \subseteq [n] \setminus I_\ell.$$

Par conséquent, $h^{\{u_1,\ldots,u_t\}}(x) = h^{\{u_1,\ldots,u_t\}}(x')$. De plus, $h(x) \neq h(x')$ car $h_{I'_\ell}(x) = x_{I_\ell} \neq {x'}_{I_\ell} = h_{I'_\ell}(x')$. Il en résulte que $\{x, x'\} \in E$ et que X_ℓ est une clique dans le graphe de confusion $G_{h,u}$. Notons que X_ℓ a une cardinalité de q^k et on obtient $\chi(G_{h,u}) \geq \omega(G_{h,u}) \geq q^k$. D'après le lemme 21, pour tout $u \in \mathbb{S}_{[n]}$, on a $\boldsymbol{\kappa}(h, u) \geq k$. Par conséquent, $\boldsymbol{\kappa}(h) \geq k$. □

En utilisant le lemme 22, on peut prouver la proposition 4 ci-dessous.

Proposition 4. *Pour tout* $q \geq 2$, $\boldsymbol{\kappa}_{n,q} \geq \lfloor n/4 \rfloor$.

Démonstration. Soit $n = 4k$, $K = [2k]$, $M = [2k+1, n]$. Clairement, un sous-ensemble $I_\ell \in \binom{K}{k}$ peut être encodé par un mot $\alpha^\ell \in \{\mathtt{0}, \mathtt{1}\}^{2k}$ tel que, pour tout $i \in K$,

$$\alpha^\ell_i = \begin{cases} \mathtt{1} & \text{si } i \in K \setminus I_\ell, \\ \mathtt{0} & \text{sinon.} \end{cases}$$

Maintenant, considérons les configurations $a^1, \ldots, a^s \in \mathbb{A}^n$ telles que, pour tout $\ell \in [s]$, $a^\ell_K = (0)^{2k}$ et $a^\ell_M = \alpha^\ell$. On peut vérifier que les ensembles $X_1, \ldots, X_s$ (définis comme dans l'énoncé du lemme 22) sont disjoints. En effet, on a $M \subseteq [n] \setminus I_\ell$ pour tout $\ell \in [s]$. Par conséquent, pour tout $x \in X_\ell$ et $x' \in X_{\ell'}$ avec $\ell \neq \ell'$,

$$x_M = a^\ell_M = \alpha^\ell \neq \alpha^{\ell'} = a^{\ell'}_M = {x'}_M.$$

D'après le lemme 22 et la proposition 2, on en conclut que $\boldsymbol{\kappa}_{n,q} \geq \lfloor n/4 \rfloor$. □

On peut voir que, même pour $q = 2$, la borne donnée dans la proposition 4 est loin d'être optimale. D'abord, il y a seulement $\binom{2k}{k}$ sous-ensemble de $[2k]$ de taille k. On pourrait les encoder dans un espace de taille $\lceil \log_q(\binom{2k}{k}) \rceil$ (au lieu d'un espace de taille $2k$). Toutefois, ceci ne fait pas une grosse différence car $\binom{2k}{k}$ est proche de 2^{2k} quand k tend vers l'infini car $\binom{2k}{k} = 2^{2k+\mathcal{O}(\sqrt{k})}$. Plus important, les coordonnées $[2k] \setminus I_\ell$ de a^ℓ ne sont pas utilisées pour encoder le sous-ensemble I_ℓ. À l'inverse, dans le lemme 23 ci-dessous, les coordonnées $[2k] \setminus I_\ell$ de a^ℓ sont utilisées pour encoder la seconde moitié de α^ℓ et par conséquent, l'intervalle $[2k+1, n]$ est de taille k. On peut vérifier que les ensembles $X_1, \dots, X_s$ ainsi définis sont disjoints et on obtient donc le lemme 23.

Lemme 23. *Pour tout $q \geq 2$, $\boldsymbol{\kappa}_{n,q} \geq \lfloor n/3 \rfloor$.*

Démonstration. Soit $n = 3k$, $K = [2k]$, $M = [2k+1, n]$, $s = \binom{2k}{k}$ et $\binom{[2k]}{k} = \{I_1, \dots, I_s\}$. Pour tout $\ell \in [s]$, soit $I'_\ell = K \setminus I_\ell$ et soit $\alpha^\ell \in \{0,1\}^{2k}$ tel que, pour tout $i \in [2k]$, $\alpha^\ell_i = \begin{cases} 1 & \text{si } i \in I'_\ell \\ 0 & \text{sinon} \end{cases}$. Définissons $a^1, \dots, a^s \in \mathbb{A}^n$ tel que

$$a^\ell_{I'_\ell} = \alpha^\ell_{[k+1,2k]},\ a^\ell_{I_\ell} = (0)^k \text{ et } a^\ell_M = \alpha^\ell_{[k]}.$$

Pour tout $\ell \in [s]$, soit $X_\ell = \{x \in \mathbb{A}^n \mid x_{[n]\setminus I_\ell} = a^\ell_{[n]\setminus I_\ell}\}$. On doit maintenant prouver que les ensembles $X_1, \dots, X_n$ sont disjoints.

Pour tout $\ell \in [s]$ et $i \in [0, 2k]$, soit $\sigma^\ell(i) = |I'_\ell \cap [i]|$ avec $[0] = \emptyset$. Notons que $\sigma^\ell(i)$ est aussi égale à $i - |I_\ell \cap [i]|$ et que, pour tout $i \in I'_\ell$, on a également $x_i = \alpha^\ell_{\sigma^\ell(i)+k}$. Pour tout $j \in [0, 2k]$, on va définir $c^{(j)}$ comme suit. Premièrement, pour tout $x \in \mathbb{A}^n$, $c^{(0)}(x) = x_{[2k+1,n]}$. Deuxièmement, pout tout $x \in \mathbb{A}^n$ et $j \in [2k]$,

$$c^{(j)}(x) = \begin{cases} b & \text{si } j \leq |b| \text{ et } b_j = 0, \\ bx_j & \text{sinon,} \end{cases} \quad \text{avec } b = c^{(j-1)}(x).$$

Pour prouver que les ensembles $X_1, \dots, X_s$ sont disjoints, il suffit de prouver que, pour tout $x \in X_\ell$, on a $c^{(2k)}(x) = \alpha^\ell$. Soit $x \in X^\ell$. Par récurrence sur j, prouvons que, pour tout $j \in [0, 2k]$,

$$c^{(j)}(x) = \alpha^\ell_{[\sigma^\ell(j)+k]}.$$

Premièrement,

$$c^{(0)}(x) = x_{[2k+1,n]} = \alpha^\ell_{[k]} = \alpha^\ell_{[\sigma^\ell(0)+k]}.$$

Deuxièmement, soit $j \in [2k]$ et supposons que $c^{(j-1)}(x) = b$ avec $b = \alpha^\ell_{[\sigma^\ell(j-1)+k]}$. On peut démontrer que $c^{(j)}(x) = \alpha^\ell_{[\sigma^\ell(j)+k]}$ comme suit.

— Si $j \leq |b|$ et $b_j = 0$ alors $c^{(j)}(x) = b$. Remarquons que $\alpha^\ell_j = b_j = 0$ et donc que $j \notin I'_\ell$.

Par conséquent,

$$\sigma^\ell(j) = |I'_\ell \cap [j]| = |I'_\ell \cap [j-1]| + |I'_\ell \cap \{j\}| = |I'_\ell \cap [j-1]| = \sigma^\ell(j-1).$$

Finalement, $c^{(j)}(x) = b = \alpha^\ell_{[\sigma^\ell(j-1)+k]} = \alpha^\ell_{[\sigma^\ell(j)+k]}$.

— Sinon, on a $c^{(j)}(x) = bx_j$. On peut vérifier que $j \in I'_\ell$ comme suit.
  — Supposons que $j \leq |b|$ et $b_j = \mathtt{1}$. Alors $\alpha^\ell_j = b_j = \mathtt{1}$ et donc $j \in I'_\ell$.
  — Maintenant, supposons que $j > |b|$. Remarquons que $|b| = \sigma^\ell(j-1) + k = j - |I_\ell \cap [j-1]| + k$. Donc, $j > j - |I_\ell \cap [j-1]| + k$ et $|I_\ell \cap [j-1]| > k$. Par ailleurs, $|I_\ell| = k$ par définition de I_ℓ. De ce fait, $I_\ell \subseteq [j-1]$, $[j,n] \subseteq I'_\ell$ et $j \in I'_\ell$.

  On conclut que,

$$\sigma^\ell(j) = |I'_\ell \cap [j]| = |I'_\ell \cap [j-1]| + |I'_\ell \cap \{j\}| = \sigma^\ell(j-1) + 1.$$

  Donc, $|c^{(j)}(x)| = |bx_j| = \sigma^\ell(j-1) + k + 1 = |\alpha^\ell_{[\sigma^\ell(j)+k]}|$. De plus, $c^{(j)}_{[\sigma^\ell(j)+k]}(x) = b = \alpha^\ell_{[\sigma^\ell(j)+k]}$ et $c^{(j)}_j(x) = x_j = \alpha^\ell_{\sigma^\ell(j)+k}$. Par conséquent, $c^{(j)}(x) = \alpha^\ell_{[\sigma^\ell(j)+k]}$.

Par récurrence, pour tout $j \in [0, 2k]$, $c^{(j)}(x) = \alpha^\ell_{[\sigma^\ell(j)+k]}$ et en particulier $c^{(2k)}(x) = \alpha^\ell$. Il en résulte que les ensembles $X^1, \ldots, X^\ell$ sont disjoints. Ainsi, d'après le lemme 22 et la proposition 2, on a $\boldsymbol{\kappa}_{n,q} \geq \lfloor n/3 \rfloor$. □

Dans le lemme 24, on va étudier le cas où $q \geq 4$. On voit bien que lorsque $q \geq 4$, on peut encoder deux valeurs $\alpha^\ell_i \in \{\mathtt{0}, \mathtt{1}\}$ dans chaque valeur $a^\ell_i \in \mathbb{A}$ avec $i \in I'_\ell$. Par conséquent, on peut encoder tout α^ℓ dans $a^\ell_{I'_\ell}$. Pour permettre un décodage non ambigu, on a seulement besoin d'encoder une « position de départ » dans un espace de taille environ $\log_q(k)$. Cette position va nous indiquer une coordonnée i judicieusement choisie dans I'_ℓ. En observant a^ℓ_i, on obtiendra une partie de α^ℓ et donc de I'_ℓ. On révèle alors une plus grande partie de α^ℓ en observant les valeurs de $a^\ell_{I'_\ell}$ nouvellement révélée. On recommence cette opération jusqu'à trouver tout I'_ℓ. On en conclut que lorsque $q \geq 4$, on peut borner inférieurement $\boldsymbol{\kappa}_{n,q}$ par environ $n/2 - \log_q(n)$. Ceci est proche de la borne supérieure donnée dans le théorème 20, qui est elle-même proche de $n/2 + \log_q(n)$.

Lemme 24. *Pour tout $q \geq 4$, $\boldsymbol{\kappa}_{n,q} \geq n/2 - \log_q(n) - \mathcal{O}(1)$.*

Démonstration. Dans cette démonstration seulement, pour simplifier grandement l'utilisation des modulos, les coordonnées d'un réseau sont indexées à partir de 0 (à la place de 1). En d'autres termes, toute configuration $x \in \mathbb{A}^n$, s'écrit $x_0 x_1 \ldots x_{n-1}$. De plus nous allons beaucoup utiliser des intervalles vides $]t,t] = [t,t[= \emptyset$ avec $t \in \mathbb{N}$. Soient $n = 2k + r$ avec $r \geq \lceil \log_q(2k) \rceil$, $K = [0, 2k[$, $M = [2k, n[$, $s = \binom{2k}{k}$ et $\binom{K}{k} = \{I_1, \ldots, I_s\}$. On définit $I'_j = K \setminus I_j$ pour tout $j \in [s]$. Pour tout $\ell \in [s]$, soit $\alpha^\ell \in \{\mathtt{0}, \mathtt{1}\}^{2k}$ tel que, pour tout $i \in K$, $\alpha_i = \mathtt{1}$ si $i \in I'_\ell$ et $\mathtt{0}$ sinon. Pour tout $I, J \subseteq K$, on définit

$$\Delta(J, I) = |J \cap I| - |J \setminus I| \text{ et } M_\ell = \min_{m \in K}(\Delta([0, m[, I'_\ell)).$$

Enfin, soit $m_\ell \in K$ tel que $\Delta([0, m_\ell[, I'_\ell) = M_\ell$.

Pour tout $m \in]-2k, 2k[$, soit π_m la fonction qui prend une coordonnée $i \in K$ (*resp.* un sous-ensemble $I \subseteq K$ ou une configuration $y \in \mathbb{A}^{2k}$) et retourne la coordonnée i (*resp.* l'ensemble I ou la configuration y) translatée circulairement de m coordonnées sur la gauche. Plus formellement, pour tout $i \in K$ et $m \in]-2k, 2k[$, on a $\pi_m(i) = i - m \mod 2k$. Pour tout $I \subseteq [2k]$, on a $\pi_m(I) = \{\pi_m(i) \mid i \in I\}$. Enfin, pour tout $y \in \mathbb{A}^{2k}$, on a

$$\pi_m(y) = y_{\pi_{-m}(0)} y_{\pi_{-m}(1)} \cdots y_{\pi_{-m}(n-1)}.$$

Remarquons que $\pi_m(y) = \begin{cases} y_{[m,n[} y_{[0,m[} & \text{si } m \in [0, 2K[, \\ y_{[n+m,n[} y_{[0,n+m[} & \text{si } m \in]-2k, 0]. \end{cases}$

Pour tout $\ell \in [s]$, notons $J_\ell = \pi_m(I_\ell)$, $J'_\ell = \pi_m(I'_\ell)$ et $\beta^\ell = \pi_m(\alpha^\ell)$ avec $m = m_\ell$.

Affirmation 8. *Soit $i \in [0, 2k]$, $\ell \in [s]$. On a $\Delta([0, i[, J'_\ell) \geq 0$.*

Démonstration. Premièrement, supposons que $m_\ell + i < 2k$. Alors, $\Delta([m_\ell, m_\ell + i[, I'_\ell) = \Delta([0, i[, J'_\ell)$. De plus,

$$\Delta([0, m_\ell + i[, I'_\ell) = \Delta([0, m_\ell[, I'_\ell) + \Delta([m_\ell, m_\ell + i[, I'_\ell) = M_\ell + \Delta([0, i[, J'_\ell).$$

Par conséquent, $\Delta([0, m_\ell + i[, J'_\ell) \geq 0$ car sinon ceci contredirait la minimalité de M_ℓ. Deuxièmement, supposons que $m_\ell + i \geq 2k$. Soit $j = m_\ell + i \mod 2k$. On a $0 \leq j \leq m_\ell$. Notons que $\Delta(K, I'_\ell) = 0$ car $|I_\ell| = |I'_\ell|$. De ce fait,

$$\Delta([0, i[, J'_\ell) = \Delta([m_\ell, 2k[, I'_\ell) + \Delta([0, j[, I'_\ell) = \Delta(K, I'_\ell) - \Delta([j, m_\ell[, I'_\ell) = -\Delta([j, m_\ell[, I'_\ell).$$

Toutefois,

$$\Delta([0, m_\ell[, I'_\ell) = \Delta([0, j[, I'_\ell) + \Delta([j, m_\ell[, I'_\ell).$$

Par conséquent,

$$\Delta([0, j[, I'_\ell) = \Delta([0, m_\ell[, I'_\ell) - \Delta([j, m_\ell[, I'_\ell) = M_\ell + \Delta([0, i[, J'_\ell).$$

Il en résulte que $\Delta([0, i[, J'_\ell) \geq 0$ car sinon ceci contredirait la minimalité de M_ℓ. Dans tous les cas on a donc, $\Delta([0, i[, J'_\ell) \geq 0$.

■

On considère maintenant deux fonctions $g : [2k] \to \mathbb{A}^{|M|}$, $g' : \mathbb{A}^{|M|} \to [2k]$ qui peuvent encoder/décoder une coordonnée $i \in K$. Cela signifie que, pour tout $m \in [2k]$ $g'(g(m)) = m$. De la même façon, on considère deux fonctions $p : \{0, 1\}^2 \to A$ et $p' : A \to \{0, 1\}^2$ pour encoder/décoder un couple de valeurs d'un mot α^ℓ. En d'autres termes, pour tout $b \in \{0, 1\}^2$, on a $p'(p(b)) = b$. Notons que de telles fonctions existent bien car $q \geq 4$. Pour tout $j \in [0, 2k]$ et $\ell \in [s]$, notons $\sigma_\ell(j) = 2|J' \cap [0, j[|$ pour tout $j \in [0, 2k]$.

Définissons $a^1, \dots, a^s \in \mathbb{A}^n$ et $d^1, \dots, d^s \in \mathbb{A}^{2k}$ comme suit. Pour tout $\ell \in [s]$,

$$\forall i \in J'_\ell,\ d^\ell_i = p(\beta^\ell_{[\sigma(i),\sigma(i)+1]}),\ d^\ell_{J_\ell} = (0)^k,\ a^\ell_K = \pi_{-m_\ell}(d^\ell) \text{ et } a^\ell_M = g(m_\ell).$$

Soit $X_\ell = \{x \in \mathbb{A}^n \mid x_{[n]\setminus I_\ell} = a^\ell_{[n]\setminus I_\ell}\}$.

Pour tout $j \in [-1, 2k[$, soit la fonction $c^{(j)}$ définie comme suit. Premièrement, pour tout $y \in \mathbb{A}^{2k}$, $c^{(-1)}(y) = \epsilon$ avec ϵ le mot vide. Deuxièmement, pour tout $y \in \mathbb{A}^{2k}$,

$$c^{(j)}(y) = \begin{cases} b & \text{if } j \le |b| \text{ et } b_j = \mathtt{0} \\ bp'(y_j) & \text{sinon} \end{cases} \text{ avec } b = c^{(j-1)}(y).$$

Soit $c : \mathbb{A}^n \to \{\mathtt{0}, \mathtt{1}\}^r$ tel que $c = \pi_{-t} \circ c^{(2k)} \circ \pi_t \circ \mathrm{pr}_K$ avec $t = g'(x_M)$. Pour prouver que les ensembles $X_1, \dots, X_s$ sont disjoints, il suffit de montrer que, pour tout $\ell \in [s]$ et $x \in X^\ell$, on a $c(x) = \alpha^\ell$.

Soit $\ell \in [s]$ et $x \in X^\ell$. On a, $g'(x_M) = g'(a_M) = g'(g(m_\ell)) = m_\ell$. Donc, il suffit de montrer que $c^{(2k-1)}(y) = \beta^\ell$ avec $y = \pi(x_K, m_\ell)$. Pour tout $\ell \in [s]$ et pour tout $i \in [0, 2k]$, soit $\sigma^\ell(i) = 2|J'_\ell \cap [0, i[\ |$.

Pour alléger la notation, écrivons simplement $\beta = \beta^\ell$, $J = J_\ell$, $J' = J'_\ell$ et $\sigma = \sigma^\ell$.

Par récurrence sur j, prouvons que, pour tout $j \in [-1, 2k[$,

$$c^{(j)}(y) = \beta_{[0,\sigma(j+1)[}.$$

Premièrement,

$$c^{(-1)}(y) = \epsilon = \beta_\emptyset = \beta_{[0,0[} = \beta_{[0,\sigma(0))[}.$$

Deuxièmement, soit $j \in [2k[$ et supposons que $c^{(j-1)}(y) = b$ avec $b = \beta_{[0,\sigma(j)[}$. On peut prouver que $c^{(j)}(y) = \beta_{[0,\sigma(j+1)[}$ comme suit.

— Si $j \le |b|$ et $b_j = \mathtt{0}$ alors $c^{(j)}(y) = b$. Remarquons que $\beta_j = b_j = \mathtt{0}$ et donc que $j \notin J'$. Par conséquent,

$$\sigma(j+1) = 2|J' \cap [0, j+1[| = 2|J' \cap [0, j[| + 2|J' \cap \{j\}| = \sigma(j-1) + 0 = \sigma(j-1).$$

Il en résulte que $c^{(j)}(y) = b = \beta_{[0,\sigma(j)[} = \beta_{[0,\sigma(j+1)[}$.

— Sinon, on a $c^{(j)}(y) = bp'(y_j)$. On peut vérifier que $j \in J'$ comme suit.

— Supposons que $j \le |b|$ et $b_j = \mathtt{1}$. On a alors, $\beta_j = b_j = \mathtt{1}$ et donc $j \in I'$.

— Maintenant, supposons que $j \ge |b|$. On a alors $b = \beta_{[0,\sigma(j)[}$. Notons que

$$|J' \cap [0, j[\ | = j - |J \cap [0, j[\ |.$$

Par conséquent,

$$|b| = \sigma(j) = 2|J' \cap [0, j[\ | = j - |J \cap [0, j[\ | + |J' \cap [0, j[\ | = \Delta([0, j[, J') + j.$$

En outre, $|b| \leq j$ et donc $\Delta([0, j[, J') + j \leq j$ et $\Delta([0, j[, J') \leq 0$. Si $j \in J$ alors

$$\Delta([0, j+1[, J') = \Delta([0, j[, J') + \Delta(\{j\}, J') = \Delta([0, j[, J') - 1 < 0$$

ce qui est impossible car, d'après l'affirmation 8, on a $\Delta([0, j+1[, J') \geq 0$. Il en résulte que $j \in J'$.

Dans tous les cas, on a $j \in J'$. Par conséquent,

$$\sigma(j+1) = 2|J' \cap [0, j+1[| = 2|J' \cap [0, j[| + 2|J' \cap \{j+1\}| = \sigma(j) + 2.$$

Donc,

$$|c^{(j)}(y)| = |bp'(y_j)| = \sigma(j) + 2 = \sigma(j+1) = |\beta_{\sigma(j+1)}|.$$

De plus,

$$(c^{(j)}(y))_{[0,\sigma(j)[} = b = \beta_{[0,\sigma(j)[},$$

et

$$(c^{(j)}(x))_{\{\sigma(j),\sigma(j)+1\}} = p'(y_j) = p'(d_j) = p'(p(\beta_{\{\sigma(j),\sigma(j)+1\}})) = \beta_{\{\sigma(j),\sigma(j)+1\}}.$$

Finalement, $c^{(j)}(y) = \beta_{[0,\sigma(j+1)[}$.

Par récurrence, pour tout $j \in [0, 2k[$, $c^{(j)}(y) = \beta_{[-1,\sigma(j+1)[}$ et en particulier $c^{(2k-1)}(y) = \beta$. Donc, pour tout $\ell \in [s]$ et $x \in X^\ell$, $c(x) = \beta^\ell$ et les ensembles $X_1, \ldots, X_\ell$ sont disjoints. En utilisant le lemme 22, on a prouvé que $\kappa_{2k+r,q} \geq k$ avec $r \geq \lceil \log_q(2k) \rceil$ et $q \geq 4$.

Considérons donc un $n \in \mathbb{N}$ quelconque et le plus grand entier k' tel que $n = 2k' + \lceil \log_q(2k) + r' \rceil$ pour un certain $r' \geq 0$. Prenons $k = n/2 - \log_q(2k') - 4$ et $r = n - k$. On peut vérifier que $2k + r \leq n$ et que $r \geq \lceil \log_q(2k) \rceil$. Donc, pour tout $n \in \mathbb{N}$, $q \geq 4$, $\kappa_{n,q} \geq n/2 - \log_q(n) - \mathcal{O}(1)$. □

On peut maintenant s'intéresser au cas particulier où $q = 3$.

Lemme 25. *Si n tend vers l'infini et que $q = 3$, alors on a $\boldsymbol{\kappa}_{n,q} \geq \frac{n}{1+2\log_3(2)} - \mathcal{O}(\sqrt{n}) \geq \frac{n}{2{,}26} - \mathcal{O}(\sqrt{n})$.*

Démonstration. On peut voir que si $n \geq 2k + \log_3(2^k) = 2k + k\log_3(2)$ on a $\boldsymbol{\kappa}_{n,q} \geq k$ en adaptant la démonstration du lemme 23. Ceci nous donne une borne inférieure pour $\boldsymbol{\kappa}_{n,q}$ d'environ $n/(2 + \log_3(2)) \approx \frac{n}{2{,}6}$.

En faisant comme ça, on ne tire pas vraiment profit des valeurs de $a^\ell_{I'_\ell}$. En effet, chaque lettre a^ℓ_i avec $i \in I'_\ell$ pourrait prendre 3 valeurs mais on s'en sert seulement pour coder une

valeur $\alpha_j^\ell \in \{0,1\}$. D'un autre côté, on n'a pas assez de place pour coder un couple de valeurs $\alpha_{\{j,j'\}}^\ell \in \{0,1\}^2$ comme on en a besoin dans la preuve du lemme 24.

Cela dit, on peut utiliser l'astuce suivante. On regroupe les coordonnées de I'_ℓ par ensemble de taille λ, où λ est un paramètre qui sera fixé plus tard. Dans chaque mot $a \in \mathbb{A}^\lambda$, on peut alors coder un mot $b \in \{0,1\}^\mu$ avec μ le plus grand entier tel que $2^\mu \leq 3^\lambda$.

En adaptant la preuve du lemme 24, on obtient l'inégalité suivante. On a $\boldsymbol{\kappa}_{n,q} \geq k$ si $n = k + \gamma + r$ avec $r = \lceil \log_3(2k) \rceil$ et $\gamma \geq \lambda + (2k\frac{\lambda}{\mu})$. On a alors affaire à un problème d'optimisation où on veut minimiser γ. Pour ceci, on doit prendre λ suffisamment grand pour que la différence entre 3^λ et 2^μ soit la plus petite possible. Mais on ne doit pas prendre λ trop grand sinon on augmente γ artificiellement. On peut cependant prendre λ croissant avec k mais négligeable devant k (par exemple $\lambda = \lfloor \sqrt{k} \rfloor$). En posant $\beta = \log_3(2)$, $\mu = \lfloor \lambda/2\beta \rfloor$ et $n = k + \gamma + r$ avec $\gamma \geq \lambda + 2\beta k$, et k maximal sous ces contraintes, on obtient $n = (1+2\beta)k + \mathcal{O}(k)$. Comme l'inégalité $\gamma \geq \lambda + (2k\frac{\lambda}{\mu})$ est alors vérifiée, on en déduit $\boldsymbol{\kappa}_{n,q} \geq k = \frac{n}{1+2\beta} - \mathcal{O}(\sqrt{n})$.

□

5.6 Complexité procédurale

Dans cette section, on étudie la relation entre le coût de séquentialisation et la *complexité procédurale* (définie ici comme dans [68]). C'est basiquement le nombre d'étapes nécessaires pour calculer une fonction h, en modifiant une coordonnée à la fois. On rappelle qu'une *instruction* est une fonction $g : \mathbb{A}^n \to \mathbb{A}^n$ qui ne modifie qu'une seule coordonnée. Autrement dit, on a $g \in F^i(n,q)$ pour un certain $i \in [n]$. On rappelle également que l'on note $\mathrm{F}^*(n,q) = \{\mathrm{F}^i(n,q) \mid i \in [n]\}$ l'ensemble des instructions sur $\mathbb{A}^n$.

Définition 10 (Complexité procédurale)**.** *La* complexité procédurale *d'un réseau $h \in \mathrm{F}(n,q)$ avec une mémoire k, (noté $\mathcal{L}(h \mid k)$), est le nombre minimum t d'instructions $g^{(1)}, \ldots, g^{(t)} \in \mathrm{F}^*(n+k,q)$ tel que* $\mathrm{pr}_{[n]} \circ g^{(t)} \circ \cdots \circ g^{(1)} = h \circ \mathrm{pr}_{[n]}$.

En particulier, la complexité procédurale avec une mémoire $k = 0$ est appelé la *complexité procédurale sans mémoire* et est notée $\mathcal{L}(h)$. Cette notion, appelée *memoryless computation* en anglais, a été introduite dans [28] et développée entre autres dans [30, 31]. Ici, à l'inverse, on va s'intéresser à la complexité procédurale avec une mémoire arbitrairement grande que l'on va noter $\mathcal{L}^*(h) = \min(\{\mathcal{L}(h \mid k) \mid k \geq 0\})$. Le théorème 22 ci-dessous montre que, pour tout réseau h on a $\mathcal{L}^*(h) = \boldsymbol{\kappa}(h) + \Omega(h)$. Ceci signifie que si notre réseau n'a aucune fonction triviale alors sa complexité procédurale est exactement équivalente à son coût de séquentialisation. De plus, ceci montre que la complexité procédurale minimum est atteinte avec une mémoire de $\boldsymbol{\kappa}(h)$.

Théorème 22. *Soit $h \in \mathrm{F}(n,q)$ et $k = \boldsymbol{\kappa}(h)$. On a $\mathcal{L}^*(h) = \mathcal{L}(h \mid k) = \Omega(h) + k$.*

Démonstration. Clairement $\mathcal{L}^*(h) \leq \mathcal{L}(h \mid k)$. Il suffit donc de prouver que $\mathcal{L}(h \mid k) \leq \Omega(h) + k$ et que $\mathcal{L}^*(h) \geq \Omega(h) + k$.

Premier sens : $\mathcal{L}(h \mid k) \leq \Omega(h) + k$. Par définition de $k = \boldsymbol{\kappa}(h)$, il existe un réseau $f \in \mathrm{F}(m, q)$ avec $m = n + k$ qui séquentialise h grâce à une permutation $w \in \mathbb{S}_{[m]}$. On a donc $\mathrm{pr}_{[n]} \circ f^{(w_m)} \circ \ldots f^{(w_1)} = h \circ \mathrm{pr}_{[n]}$. Pour tout $i \in [m]$, $f^{w_i} \in F^*(m, q)$ et à ce stade, on a déjà prouvé que $\mathcal{L}^*(h) \leq \mathcal{L}(h \mid k) \leq k + n$.

Mais s'il existe une fonction triviale h_i, alors f^{w_j} avec $w_j = i$ est la fonction identité. Si on considère $v \in [m]^{k+\Omega(h)}$, le sous-mot de w auquel on a retiré toutes les coordonnées i telles que $h^{(i)}$ est l'identité, on voit que $f^v = f^w$. Le réseau h peut donc être calculé par $\Omega(h) + k$ instructions dans $F^*(m, q)$ ce qui nous donne le résultat désiré.

Deuxième sens : $\mathcal{L}^*(h) \geq \Omega(h) + \boldsymbol{\kappa}(h)$

Notons $\ell = \mathcal{L}^*(h)$, $m \leq n$ et $g^{(1)}, \ldots, g^{(\ell)} \in \mathrm{F}^*(m, q)$ l'ensemble des instructions telles que

$$\mathrm{pr}_{[n]} \circ g^{(\ell)} \circ \cdots \circ g^{(1)} = h \circ \mathrm{pr}_{[n]}\,.$$

Toute fonction $g^{(i)}$ (avec $i \in [\ell]$) met à jour une coordonnée. En effet, sinon, $g^{(i)}$ serait la fonction identité, que l'on pourrait retirer de notre liste d'instructions. Ceci contredirait la définition de ℓ. Soit $w \in [m]^\ell$ un mot tel que, pour tout $i \in [\ell]$, w_i est la coordonnée mise à jour par $g^{(i)}$. Soit I l'ensemble des coordonnées de $[n]$ mise à jour par w. Soit $T = \{t_1, t_2, \ldots, t_p\} \subseteq [\ell]$ avec $t_1 < t_2 < \cdots < t_p$ l'ensemble des étapes de temps auxquelles une coordonnée de I est mise à jour pour la dernière fois par w. Plus formellement, pour tout $t \in T$, $w_t \in [n]$ et pour tout $t' \in [t+1, \ell]$, $w_t \neq w_{t'}$. De plus, prenons $S = \{s_1, s_2, \ldots, s_k\} = [\ell] \setminus T$ avec $s_1 < s_2 < \cdots < s_k$ l'ensemble des autres étapes de temps.

Posons $k = \ell - p$. On sait que $\Omega(h) \geq p$, car pour calculer h, chaque coordonnée i telle que h_i est non triviale doit être mise à jour au moins une fois. Donc $\mathcal{L}^*(h) \geq \Omega(h) + k$. Il ne reste plus qu'à prouver que $k \geq \boldsymbol{\kappa}(h)$.

Prenons $v \in \mathbb{S}_{[n]\setminus I}$ une permutation quelconque des coordonnées de $[n]$ non mises à jour dans w. Et prenons $u = (w_{t_1}, w_{t_2}, \ldots, w_{t_p})v \in \mathbb{S}_{[n]}$. On va prouver que h peut se séquentialiser dans l'ordre u avec une mémoire additionnelle de k.

Pour ça, on va s'intéresser au graphe de confusion $G_{h,u} = (\mathbb{A}^n, E)$. On définit la fonction de coloration $c : \mathbb{A}^n \to \mathbb{A}^k$ suivante. Pour tout $i \in [k]$,

$$c_i : x \mapsto g^{(s_i)}_{w_{s_i}} \circ g^{(s_i-1)} \circ \ldots \circ g^{(1)}(xy),$$

avec $y = (\mathtt{0})^{m-n}$. Prouvons que c est une coloration valide de $G_{h,u}$. Soit $\{x, x'\} \in E$. Par l'absurde, supposons que $c(x) = c(x')$. Soit $z = xy$ et $z' = x'y$. Soit i^* la plus grande coordonnée $i \in [n]$ telle que $h^{\{u_1,\ldots,u_i\}}(x) = h^{\{u_1,\ldots,u_i\}}(x')$. Soit $t \in T$ la dernière étape de temps à laquelle la coordonnée u_{i^*} est mise à jour. Soit $r = g^{(t)} \circ \cdots \circ g^{(1)}(z)$ et

$r' = g^{(t)} \circ \cdots \circ g^{(1)}(z')$. Prouvons que $r = r'$. Pour toute coordonnée $j \in [m]$ qui n'est pas encore mise à jour par w à l'étape de temps t :

— Si $j \in [n]$ alors $r_j = x_j = {x'}_j = r'_j$ car $h^{\{u_1,\dots,u_{i^*}\}}(x) = h^{\{u_1,\dots,u_{i^*}\}}(x')$, $[n] \setminus \{w_1,\dots,w_t\} = [n] \setminus \{u_1,\dots,u_{i^*}\}$ et donc

$$x_{[n]\setminus\{w_1,\dots,w_t\}} = x_{[n]\setminus\{u_1,\dots,u_{i^*}\}} = {x'}_{[n]\setminus\{u_1,\dots,u_{i^*}\}} = {x'}_{[n]\setminus\{w_1,\dots,w_t\}}.$$

— Si $j \in [n+1, m]$ alors $r_j = z_j = y_{j-n} = z'_j = r'_j$.

Pour toute coordonnée $j \in [m]$ qui n'est pas déjà mise à jour par w à l'étape de temps t :

— Si $t \in T$ alors $r_j = h_j(x) = h_j(x') = r'_j$ car $h^{\{u_1,\dots,u_{i^*}\}}(x) = h^{\{u_1,\dots,u_{i^*}\}}(x')$, $[n] \cap \{w_1,\dots,w_t\} = [n] \cap \{u_1,\dots,u_{i^*}\}$ et $j \in \{u_1,\dots,u_{i^*}\}$.

— Sinon, soit $t' < t$ la dernière étape dans w avant t telle que la coordonnée j a été mise à jour. En d'autres termes, $w_{t'} = j$ et $\forall s \in [t'+1, t[, w_s \neq j$. Soit α tel que $s_\alpha = t'$. On a $r_j = g_j^{(t')} \circ g^{(t'-1)} \circ \cdots \circ g^{(1)}(z) = c_\alpha(x) = c_\alpha(x') = g_j^{(t')} \circ g^{(t'-1)} \circ \cdots \circ g^{(1)}(z') = r'_j$.

On a donc $g^{(t)} \circ \cdots \circ g^{(1)}(z) = g^{(t)} \circ \cdots \circ g^{(1)}(z')$ et donc $g^{(\ell)} \circ \cdots \circ g^{(1)}(z) = g^{(\ell)} \circ \cdots \circ g^{(1)}(z')$. Par conséquent,

$$pr_{[n]} \circ g^{(\ell)} \circ \cdots \circ g^{(1)}(z) = h(x) \neq h(x') \neq g^{(\ell)} \circ \cdots \circ g^{(1)}(z').$$

C'est absurde, donc pour toute arête $\{x, x'\} \in E$, on a $c(x) \neq c(x')$. Donc, c est une coloration valide de $G_{h,u}$. De plus, elle utilise au plus $q^k = q^{\ell-p} \leq q^{\mathcal{L}^*(h)-\Omega(h)}$ couleurs. D'après le lemme 21, $\boldsymbol{\kappa}(h,u) \leq \mathcal{L}^*(h) - \Omega(h)$ et donc $\Omega(h) + k \leq \mathcal{L}^*(h)$.

□

Dans l'article [68], la proposition 12 affirme que, pour tout réseau $h \in \mathrm{F}(n,q)$, on a $\mathcal{L}(h \mid n-1) \leq 2n-1$. Dans le corolaire 8 ci-dessous, on améliore cette borne grâce aux théorèmes 20 et 22 et grâce au fait que, pour tout réseau $h \in \mathrm{F}(n,q)$, on a $\Omega(h) \leq n$.

Corolaire 8. *Pour tout $h \in \mathrm{F}(n,q)$, $\mathcal{L}(h \mid k) \leq k$ avec $k := \lceil n/2 + \log_q(n/2+1) \rceil$.*

De plus, dans le Corolaire 9 ci-dessous, on donne une borne inférieure à la complexité procédurale avec une mémoire arbitrairement grande. C'est une conséquence directe du théorème 22, du lemme 22 et du théorème 21.

Corolaire 9. *Pour tout $n, q \geq 2$, il existe un réseau $h \in \mathrm{F}(n,q)$ tel que $\mathcal{L}^*(h) \geq n + \lfloor n/3 \rfloor$. De plus, si $q \geq 4$, alors il existe un réseau $h \in \mathrm{F}(n,q)$ tel que $\mathcal{L}^*(h) \geq n + \lfloor n/2 - \log_q(n) \rfloor$.*

5.7 Graphe d'interaction

Dans cette section, on va montrer que l'on peut borner le coût de séquentialisation d'un réseau grâce à son graphe d'interaction.

Prenons $G = (V, A)$, un graphe d'interaction et $h \in \mathrm{F}(G, q)$. En guise d'échauffement, on peut se convaincre que si G est acyclique, alors $\boldsymbol{\kappa}(h) = 0$. En effet, on peut considérer un ordre topologique $u \in \mathbb{S}_V$ sur les sommets de G. Il n'existe alors aucun arc $(i, j) \in A$ avec $u(i) < u(j)$. Par conséquent, pour tout $x \in \mathbb{A}^n$, et $i \in V$, notons $t = u(i)$, on a

$$h_i(h^{(u_1,\dots,u_{t-1})}(x)) = h_i(x).$$

Ceci signifie que h peut se séquentialiser lui-même grâce à u : on a $h^u = h$.

Plus généralement, on peut voir que connaitre un *feedback vertex set* (FVS) de G (qui se traduit par « Coupe-cycles de sommets » en français) nous donne une borne supérieure pour $\boldsymbol{\kappa}(h)$. Le FVS d'un graphe (dirigé ou non) est un sous-ensemble de ses sommets qui, une fois retirés du graphe, le rend acyclique. En particulier, quand G est acyclique, il a un *feedback vertex set* de taille 0.

Proposition 10. *Considérons un digraphe G avec un feedback vertex set I de taille k. Pour tout réseau $h \in \mathrm{F}(G, q)$, on a $\boldsymbol{\kappa}(h) \leq k$.*

Démonstration. Notons $m = n + k$ et $M = [n + 1, m]$. On considère la fonction $f \in \mathrm{F}(m, q)$ suivante. Pour tout $x \in \mathbb{A}^n$ et $y \in \mathbb{A}^k$. On prend

$$f_M(xy) = h_I(x),\ f_I(xy) = y \text{ et } f_{V\setminus I}(xy) = h_{V\setminus I}(x)$$

Considérons $u \in \mathbb{S}_{V\setminus I}$ un ordre topologique sur $V \setminus I$ et deux permutations quelconques $v \in \mathbb{S}_M$ et $v' \in \mathbb{S}_I$. Prenons $w = vuv'$. Par simple expansion de f^w on obtient

$$xy \xrightarrow{f^v} xh_I(x) \xrightarrow{f^u} h^{V\setminus I}(x)h_I(x) \xrightarrow{f^{v'}} h(x)h_I(x),$$

et on a donc $\mathrm{pr}_{[n]} \circ f^w = h \circ \mathrm{pr}_{[n]}$ et donc $\boldsymbol{\kappa}(h) \leq k$. □

On va définir une notion plus précise que le FVS : la *directed vertex separation* (DVS) que l'on pourrait traduire en « séparation dirigée en sommet ». Cette notion a été introduite en 2008 [162] et est une généralisation de la notion de la *vertex separation* pour des graphes non dirigés. Cette notion-ci a été introduite en 1994 [55].

On en donne ici une définition adaptée aux notations de cette thèse.

Définition 11 (Directed vertex separation (DVS))**.** *Étant donné un digraphe $G = (V, A)$ et une permutation $u \in \mathbb{S}_{[n]}$, soit*

$$V_{G,u}(t) := \{i \in V \mid \exists (j, i) \in A \text{ tel que } u(i) \leq t < u(j)\}.$$

On définit la DVS dans l'ordre u notée $\mathrm{dvs}_u(G)$ *comme suit :*

$$\mathrm{dvs}_u(G) = \max_{t\in[n]} |V_{G,u}(t)|.$$

La DVS, notée $\mathrm{dvs}(G)$ *peut être définie par* $\mathrm{dvs}_u(G) = \min_{u \in \mathbb{S}_V} \mathrm{dvs}_u(G)$.

Pour tout $u \in \mathbb{S}_{[n]}$, on note $\tilde{u} := u_n u_{n-1} \dots u_1 \in \mathbb{S}_{[n]}$ le mot miroir de u.

Théorème 23. *Considérons un digraphe* $G = (V, A)$ *et un réseau* $h \in \mathrm{F}(G, q)$. *Pour tout* $u \in \mathbb{S}_{[n]}$ *on a* $\boldsymbol{\kappa}(h, u) \leq \mathrm{dvs}_{\tilde{u}}(G)$ *et donc* $\boldsymbol{\kappa}_h \leq \mathrm{dvs}(G)$.

Démonstration. Considérons une permutation $u \in \mathbb{S}_{[n]}$. On a

$$\begin{aligned} V_{G,\tilde{u}}(t) &:= \{i \in V \mid \exists (j,i) \in A \text{ tel que } \tilde{u}(i) \leq t < \tilde{u}(j)\} \\ &= \{i \in V \mid \exists (j,i) \in A \text{ tel que } u(j) < t \leq u(i)\}. \end{aligned}$$

Pour alléger la notation, on va écrire $d = \mathrm{dvs}_{\tilde{u}}(G)$ et $V(t) = V_{G,\tilde{u}}(t)$. Pour tout $i \in V$ soit $T(i) = \{t \in [n] \mid i \in V(t)\}$. De plus, quand $T(i)$ n'est pas vide, on définit $v(i) = \min(T(i))$.

Affirmation 9. *Pour tout* $i \in V$, $T(i)$ *est soit vide, soit c'est un intervalle* $[v(i), u(i)]$.

Démonstration. Supposons que $T(i)$ n'est pas vide. Par définition de $V(t)$, pour tout $t \in T(i)$, il existe donc un arc $(j, i) \in A$ tel que $u(j) < t \leq u(i)$. On a donc $T(i) \subseteq [v(i), u(i)]$.

Maintenant considérons la coordonnée j qui permet à $v(i)$ d'appartenir à $V(i)$. Il existe donc un arc $(j, i) \in A$ tel que $u(j) < v(i) \leq u(i)$. Clairement, pour tout $t \in [v(i), u(i)]$, on a $u(j) < v(i) \leq t \leq u(i)$ et donc $t \in T(i)$. En conclusion, $T(i) = [v(i), u(i)]$.

■

On va définir la coloration $c : V \mapsto [d] \cup \{*\}$ suivante. Premièrement, pour toute coordonnée i telle que $T(i) = \emptyset$, $c(i) = *$. Notons, $C_* = \{i \in V \mid c(i) = *\}$. Ensuite, on va choisir les couleurs des coordonnées $i \in V \setminus C_*$ par ordre croissant de v. Chaque couleur $c(i)$ est définie comme la plus petite couleur non attribuée à une coordonnée de $j \in V(v(i))$ déjà définie. Autrement dit, pour tout $i \in V \setminus C_*$ et $j \in V(v(i))$, on a $c(i) \neq c(j)$. Remarquons que cette définition de $c(i)$ est valable car, par définition de d, on a $|V(v(i))| \leq d$. Pour tout $i \in V$, on définit $C(i) = \{j \in V \mid c(i) = c(j)\}$.

Affirmation 10. *Pour tout* $i, j \in V$ *distincts, si* $c(i) = c(j) \neq *$ *alors* $T(i) \cap T(j) = \emptyset$.

Démonstration. Prenons $v(i) < v(j)$. Supposons qu'il existe $t \in T(i) \cap T(j)$. On a donc $v(i) \leq v(j) \leq t \leq u(i)$. Par l'affirmation précédente, comme $v(j) \in [v(i), u(i)]$, on a $i \in V(v(i))$. Par conséquent, on ne peut pas avoir $c(i) = c(j)$.

■

Affirmation 11. *Soit* $i \in V \setminus C_*$. *Pour tout* $j \in C(i)$, *avec* $u(i) < u(j)$, *on a la propriété suivante. Pour tout* $k \in \mathcal{N}(j)$, $u(i) \leq u(k)$.

Démonstration. Prenons $i, j, k \in V$ avec $c(i) = c(j) \neq *$, $k \in \mathcal{N}(j)$ et $u(i) < u(j)$. Prouvons que $u(i) \leq u(k)$. Comme $i, j \notin C_*$, $T(i)$ et $T(j)$ ne sont pas vides et $v(i)$ et $v(j)$ sont bien définis. On peut voir que $u(i) < v(j)$. En effet, sinon on aurait $u(i) \in [v(j), u(j)]$, $u(i) \in T(j)$ et, par l'affirmation précédente, $c(i) \neq c(j)$. Comme $(k, j) \in A$, $[u(k) + 1, u(j)] \subseteq T(j)$. Par conséquent, si $u(k) < u(i)$ alors $v(i) \leq u(i)$. Ceci est absurde comme on l'a déjà montré. Donc $u(i) \leq u(k)$.

■

Soit $m = n + d$, et $M = [n + 1, m]$. Construisons maintenant notre fonction $f \in \mathrm{F}(m, q)$ qui séquentialise h dans l'ordre u.

Pour tout $x \in \mathbb{A}^n$ et $y \in \mathbb{A}^d$, on définit $f(xy)$ comme suit.

— Pour tout $\ell \in [d]$, $f_{n+\ell}(xy) = \sum_{i \in C_\ell} x_\ell$.

— Pour tout $\ell \in [d]$, $i \in C_\ell$,

$$f_i(xy) = y_\ell - \sum_{\substack{j \in C_\ell \\ u(j) < u(i)}} x_j - \sum_{\substack{j \in C_\ell \\ u(i) < u(j)}} h_j(x).$$

— Pour tout $i \in C_*$, $f_i(xy) = h_i(x)$.

Toutes les sommes et soustractions sont bien sûr faites modulo q.

On peut considérer une permutation $w = (n + 1, \ldots, m)u \in \mathbb{S}_{[m]}$ qui met d'abord à jour M et ensuite $[n]$ dans l'ordre u.

Considérons $x \in \mathbb{A}^n$ et $y \in \mathbb{A}^d$. Notons $y' \in \mathbb{A}^d$, la configuration telle que, pour tout $\ell \in d$, $y'_\ell = \sum_{i \in C_\ell} x_\ell$. Montrons que $f^w(xy) = h(x)y'$. Pour ça, montrons par récurrence que, pour tout $t \in [0, n]$,

$$f^{(w_1, \ldots, w_{d+t})}(xy) = h^{\{u_1, \ldots, u_t\}}(x)y'$$

D'abord, pour $t = 0$, on a bien $f^{(w_1, \ldots, w_d)}(xy) = xy'$. Ensuite, pour $t \in [n]$, notons $x' = h^{\{u_1, \ldots, u_{t-1}\}}(x)$, et supposons que $f^{(w_1, \ldots, w_{d+t-1})}(xy) = x'y'$. Montrons que, $f_{w_{d+t}}(x'y') = h_{u_t}(x)$. Soit $i = u_t$ (*i.e.* $t = u(i)$).

Supposons d'abord que $i \in C_*$. On a $f_i(x'y') = h_i(x')$. Comme $i \in C_*$, on sait que $T(i)$ est vide et que $i \notin V(t)$. Par conséquent, il n'existe pas d'arc $(j, i) \in A$ tel que $u(j) < u(i)$. Donc ${x'}_{\mathcal{N}(i)} = x_{\mathcal{N}(i)}$ et $f_i(x'y') = h_i(x') = h_i(x)$.

Supposons maintenant que $i \in C_\ell$ avec $\ell \in [d]$. On a

$$f_i(x'y') = {y'}_\ell - \sum_{\substack{j \in C_\ell \\ u(j) < u(i)}} {x'}_j - \sum_{\substack{j \in C_\ell \\ u(i) < u(j)}} h_j(x').$$

Premièrement, on sait que ${y'}_\ell = \sum_{j \in C_\ell} h_j(x)$. Deuxièmement, pour tout $j \in C_\ell$ avec $u(j) <$

$u(i)$, on a ${x'}_j = h_j(x)$. Troisièmement et dernièrement, pour tout $j \in C_\ell$ avec $u(i) < u(j)$, on a $h_j(x') = h_j(x)$. En effet, d'après l'Affirmation 11, pour tout $k \in \mathcal{N}(j)$, $u(i) \leq u(k)$. Donc ${x'}_{\mathcal{N}(j)} = x_{\mathcal{N}(j)}$ et $h_j(x') = h_j(x)$.

Par conséquent, on a

$$f_i(x'y') = \sum_{j \in C_\ell} h_j(x) - \sum_{\substack{j \in C_\ell \\ u(j) < u(i)}} h_j(x) - \sum_{\substack{j \in C_\ell \\ u(i) < u(j)}} h_j(x) = \sum_{\substack{j \in C_\ell \\ u(j) = u(i)}} h_j(x) = h_i(x).$$

On a donc $f^{(w_1,\dots,w_{d+t})}(xy) = h^{\{u_1,\dots,u_t\}}(x)y'$ et par récurrence $f^w(xy) = h(x)y'$. Par conséquent, $\boldsymbol{\kappa}(h,u) \leq \mathrm{dvs}_{\tilde{u}}(G)$ et $\boldsymbol{\kappa}(h) \leq \mathrm{dvs}(G)$. □

Introduisons un dernier nombre : le *directed pathwidth* d'un digraphe G. On pourrait traduire cet terme par la « largeur en chemin dirigé » et allons le définir ci-dessous.

Définition 12 (Décomposition en chemin dirigé)**.** *Étant donné un digraphe $G = (V, A)$, une* décomposition dirigée en chemin *(*directed path decomposition *en anglais) est un ensemble d'ensembles $V_1, \dots, V_p \subseteq V$ tel que*

- *Pour tout $\bigcup_{i \in V} V_i = V$.*
- *Pour tout $1 \leq i \leq j \leq k \leq p$, $V_i \cap V_k \subseteq V_j$.*
- *Pour tout $(i,j) \in A$, la condition suivante est respectée. Il existe deux étapes de temps $t \in [p]$ et $t' \in [p]$ telles que $t \leq t'$ et $i \in V_t$ et $j \in V_{t'}$.*

On appelle *directed pathwidth* du digraphe G, et on note $\mathrm{dpw}(G)$ la cardinalité du plus grand ensemble V_t moins 1. Il se trouve que, pour tout digraphe G, on a $\mathrm{dpw}(G) = \mathrm{dvs}(G)$. (voir le théorème 4.4 de l'article [162]). On en déduit immédiatement le corolaire suivant.

Corolaire 10. *Considérons un digraphe $G = (V, A)$ et un réseau $h \in \mathrm{F}(G, q)$. Pour tout $u \in \mathbb{S}_{[n]}$ on a $\boldsymbol{\kappa}(h) \leq \mathrm{dpw}(G)$.*

5.8 Conclusion et perspectives

Dans ce chapitre, on a étudié le coût de séquentialisation.

On a vu que, pour tout $q \geq 4$, on pouvait encadrer le coût de séquentialisation de la manière suivante.

$$\lfloor n/2 - \log_q(n) \rfloor \leq \boldsymbol{\kappa}_{n,q} \leq \tilde{\boldsymbol{\kappa}}_{n,q} \leq \lceil n/2 + \log_q(n/2+1) \rceil.$$

Donc, pour tout n fixé, $\tilde{\boldsymbol{\kappa}}_{n,q}$ et $\boldsymbol{\kappa}_{n,q}$ tendent vers $n/2$ quand q tend ver l'infini. C'est un argument en faveur de la conjecture qui dit que, pour tout n et q on a $\tilde{\boldsymbol{\kappa}}_{n,q} = \boldsymbol{\kappa}_{n,q} = \lfloor n/2 \rfloor$.

On a aussi montré que le graphe d'interaction d'un réseau pouvait donner des bornes sur son coût de séquentialisation maximum.

L'évolution la plus naturelle en partant de cette question va être étudiée dans le chapitre suivant. À savoir, on va s'autoriser à mettre à jour plusieurs fois la même coordonnée entre deux étapes de temps et cela va changer profondément le problème.

Un autre type de question porte sur la possibilité d'ajouter des contraintes soit sur le réseau h qui doit être simulé, mais aussi sur les réseaux f qui doivent le simuler. En effet, les réseaux f ont souvent un graphe complet ou quasi-complet.

Chapitre 6

Calcul séquentiel avec répétition

6.1 Introduction

Ce chapitre porte entièrement sur la famille des modes de mise à jour séquentiels avec répétition. Ceux-ci diffèrent des modes de mise à jour séquentiels classiques car une même coordonnée peut être mise à jour un nombre arbitraire de fois entre deux étapes de temps et n'est donc pas bloc-séquentiel en général. Un mode de mise à jour séquentiel avec répétition pour un réseau $f \in \mathrm{F}(n, q)$ peut être décrit par un mot fini quelconque (éventuellement vide) sur l'alphabet $[n]$.

Le *calcul sans mémoire* est un champ de recherche qui consiste à calculer une transformation de $\mathbb{A}^n$ en exécutant une série d'instructions, c'est-à-dire en ne touchant qu'une coordonnée à la fois et sans faire appel à une mémoire supplémentaire. L'idée principale a été introduite dans [28, 29, 32, 33, 34, 35], et étendue dans [38, 39, 68]. L'exemple de base a déjà été présenté dans le chapitre précédent. Il s'agit de la fonction $h : x_1x_2 \mapsto x_2x_1$ que l'on peut simuler en exécutant la séquence d'instructions suivantes :

$$x_1x_2 \mapsto (x_1 + x_2, x_2),\ x_1x_2 \mapsto (x_1, x_1 + x_2) \text{ et } x_1x_2 \mapsto (x_1 + x_2, x_2).$$

Dans la plupart des travaux sur le calcul sans mémoire, différentes instructions peuvent être utilisées pour mettre à jour une même coordonnée. En ce qui nous concerne, nous souhaitons calculer des fonctions à l'aide d'un réseau f fixé et d'un mode séquentiel avec répétition. De ce fait, la seule instruction que l'on peut utiliser pour mettre à jour la coordonnée i est $f^{(i)}$.

Dans ce chapitre, la section 6.2 est consacrée aux définitions et notations.

Dans la section 6.3, on montre que l'on peut construire des réseaux dits n-complets (très grands) qui peuvent simuler tous les réseaux en un temps $2n$ (*i.e.* avec un modes de mise à jour séquentiels avec répétition w de taille $2n$). On ne prouve pas que c'est le temps minimum, mais on donne une borne inférieure en utilisant des résultats démontrés dans le chapitre précédent.

Dans la section 6.4, on s'intéresse à la taille minimum d'un réseau n-complet. On montre que l'on peut toujours créer un réseau n-complet de taille $n + 1$ et que c'est le minimum.

Dans la section 6.5, on généralise un peu le problème et on parle de réseaux n, q-complets. Un réseau est n, q-complet s'il peut simuler tous les réseaux de $\mathrm{F}(n, q)$. Il peut lui-même travailler sur un alphabet plus grand. On définit deux notions de simulation qui semblent bien s'appliquer au problème. On montre qu'en utilisant l'une d'elles on peut construire des réseaux n, q-complets dans $\mathrm{F}(n, 2q)$. Avec l'autre définition, on en trouve dans $\mathrm{F}(n, q + 1)$ quand n est suffisamment grand par rapport à q.

Enfin, dans la section 6.6, on ne cherche plus des réseaux f qui simulent un grand nombre de fonctions. À l'inverse, on part d'un réseau $h \in \mathrm{F}(n, q)$ et on cherche à savoir

s'il est simulable sans mémoire additionnelle. En d'autres mots, on veut savoir s'il existe un réseau $f \in \mathrm{F}(n,q)$ qui le simule. On prouve que, sauf quand l'alphabet est booléen, il existe des réseaux de toutes tailles qui sont non simulables sans mémoire supplémentaire.

Le travail présenté dans les sections 6.2 et 6.3 de chapitre ont fait l'objet d'un papier qui est en cour de relecture [24]. Les sections 6.5 et 6.6 feront l'objet d'un futur papier.

6.2 Définitions et notations

Considérons un ensemble de fonctions $g^{\langle 1\rangle}, \dots, g^{\langle p\rangle} \in \mathrm{F}(n,q)$. Notons $\langle g^{\langle 1\rangle}, \dots, g^{\langle p\rangle}\rangle$ l'ensemble des fonctions $h \in \mathrm{F}(n,q)$ que l'on peut obtenir en composant les fonctions $g^{\langle 1\rangle}, \dots, g^{\langle p\rangle}$. On s'autorise à utiliser une même fonction le nombre de fois que l'on veut et à les associer dans n'importe quel ordre. Autrement dit,

$$\langle g^{\langle 1\rangle}, \dots, g^{\langle p\rangle}\rangle := \{g^{\langle u_1\rangle} \circ \cdots \circ g^{\langle u_t\rangle} \mid t \in \mathbb{N} \text{ et } u \in [p]^t\}.$$

La fonction identité (id) appartient toujours à $H = \langle g^{\langle 1\rangle}, \dots, g^{\langle p\rangle}\rangle$ car $f^u = \mathrm{id}$ en choisissant u comme le mot vide. On dit que les fonctions $g^{\langle 1\rangle}, \dots, g^{\langle p\rangle}$ génèrent H.

Maintenant, pour tout $f \in \mathrm{F}(m,q)$, on note

$$S_f := \langle f^{(1)}, \dots, f^{(m)}\rangle.$$

Autrement dit, S_f est l'ensemble des fonctions calculables par f avec une mise à jour séquentielle intégrant éventuellement des répétitions. On a donc $S_f = \{f^w \mid w \in [m]^*\}$.

Définition 13 (Simulation séquentielle)**.** *Posons $m \geq n$. On dit que $f \in \mathrm{F}(m,q)$* simule séquentiellement *$h \in \mathrm{F}(n,q)$ s'il existe un mot $w \in [m]^*$ tel que*

$$\mathrm{pr}_{[n]} \circ f^w = h \circ \mathrm{pr}_{[n]}\,.$$

Le temps de simulation*, noté $\mathrm{t}_f(h)$ correspond à la taille du plus petit tel mot w.*

Cela signifie que si $f \in \mathrm{F}(m,q)$ simule séquentiellement $h \in \mathrm{F}(n,q)$ à l'aide de $w \in [m]^*$ alors, pour tout $x \in \mathbb{A}^n$ et $y \in A^{m-n}$, on a $f^w_{[n]}(xy) = h(x)$.

De manière totalement équivalente, on peut dire que le réseau $f \in \mathrm{F}(m,q)$ simule séquentiellement h s'il existe un réseau $g \in \mathrm{S}_f$ tel que $\mathrm{pr}_{[n]} \circ g = h \circ \mathrm{pr}_{[n]}$. Le *temps de simulation* $\mathrm{t}_f(h)$ correspond alors à la complexité procédurale de h en utilisant uniquement des fonctions de $\{f^{(1)}, \dots, f^{(m)}\}$.

Définition 14 (n-complétude)**.** *Soit $m \geq n$. Un réseau $f \in \mathrm{F}(m,q)$ est* n-complet*, s'il simule tous les réseaux $h \in \mathrm{F}(n,q)$. On peut alors définir le temps de simulation général de*

f comme

$$t_f(n) := \max\{t_f(h) \mid h \in \mathrm{F}(n,q)\}.$$

Dans ce chapitre, pour définir une instruction $g \in \mathrm{F}^i(n,q)$ à partir d'une fonction locale $h : \mathbb{A}^n \to \mathbb{A}$ on notera $g : x_i \leftarrow h(x)$ pour signifier que $g : x \mapsto y$ avec $y_i = h(x)$ et $y_{[n]\setminus\{i\}} = x_{[n]\setminus\{i\}}$.

6.3 Réseau n-complet de temps minimum

Commençons par introduire un outil de base de nos constructions : le *commutateur*. Ce dernier nous permet d'encoder et de lire un bit d'information (ou, plus généralement, une lettre dans $\mathbb{A}$). Si on connaissait la configuration initiale pour une certaine coordonnée i (disons $x_i = 0$), alors il serait très facile de construire un commutateur à l'aide d'une simple fonction locale : le commutateur est allumé si $x_i \mod 2 = 1$ et éteint si $x_i \mod 2 = 0$. On pourrait utiliser l'instruction $f^{(i)} : x_i \leftarrow (x_i + 1) \mod 2$ pour allumer le commutateur et l'exécuter de nouveau pour l'éteindre. Cependant, en général, on ne connait pas l'état initial de notre configuration. On va donc utiliser deux coordonnées i et j et définir,

$$f^{(i)} : x_i \leftarrow x_j \text{ et } f^{(j)} : x_j \leftarrow x_i + 1.$$

Dans le cas présent, on peut dire que le commutateur est allumé si $x_i = x_j$ et éteint si $x_i \neq x_j$. L'instruction $f^{(i)}$ allume le commutateur, alors que l'instruction $f^{(j)}$ l'éteint. On peut aussi créer un $\mathbb{A}$-commutateur qui encode une valeur dans $\mathbb{A}$. Pour cela, on utilise q coordonnées $i_0, \ldots, i_{q-1}$ et, pour tout $a \in \mathbb{A}$, on définit,

$$f^{(i_a)} : x_{i_a} \leftarrow a - \sum_{b \in \mathbb{A}\setminus\{a\}} x_{i_b}.$$

La valeur du $\mathbb{A}$-commutateur est le résultat de la somme $\sum_{b\in\mathbb{A}} x_{i_b}$. On peut vérifier que l'instruction $f^{(i_a)}$ passe alors la valeur du commutateur à a. Dans le théorème 24 ci-dessous, on va voir une variation de cette idée de commutateur. On va s'en servir pour créer un réseau n-complet en temps $2n$.

Théorème 24. *Pour tout entier $n \geq 1$ et $q \geq 2$, il existe $f \in \mathrm{F}(sn, q)$ n-complet en temps $2n$ avec $s = q^{q^n}$.*

Démonstration. Pour tout $i \in [n]$, énumérons les instructions de $\mathrm{F}^i(n,q)$ (privé de l'identité) comme

$$g^{\langle i,1\rangle}, \ldots, g^{\langle i,s-1\rangle}.$$

Posons $m = sn$. Nous allons également désigner par $\langle i, \ell\rangle$ chaque coordonnée de l'ensemble $[n+1, m]$. L'entier $i \in [n]$ désigne le numéro du commutateur et $\ell \in [s-1]$ l'indice de l'une

de ses coordonnées. Définissons maintenant $f \in \mathrm{F}(m,q)$. D'abord, pour tout $i \in [n]$, posons

$$f^{(i)} : x_i \leftarrow \sum_{\ell \in [s-1]} x_{\langle i,\ell \rangle}.$$

Chaque instruction $f^{(i)}$ place dans la i-ème composante du réseau la valeur du commutateur i. Ensuite, pour tout $i \in [n]$ et $\ell \in [s-1]$, posons

$$f^{(\langle i,\ell \rangle)} : x_{\langle i,\ell \rangle} \leftarrow g_i^{\langle i,\ell \rangle}(x_{[n]}) - \sum_{j \in [s-1] \setminus \{\ell\}} x_{\langle i,j \rangle}.$$

Autrement dit, exécuter $f^{(\langle i,\ell \rangle)}$ met la valeur du commutateur i à $g_i^{\langle i,\ell \rangle}(x_{[n]})$.

Considérons une fonction $h \in \mathrm{F}(n,q)$. Quand $\Omega(h) = n$ (*i.e.* h n'a aucune fonction locale triviale), on peut voir chaque instruction $h^{(i)}$ comme une instruction $g^{\langle i,\ell_i \rangle}$.

On peut calculer cette fonction comme suit :

Étape 1. pour chaque i entre 1 et n, si $h^{(i)}$ n'est pas la fonction identité, on donne la valeur $h_i(x)$ au i-ème commutateur en exécutant $f^{(\langle i,\ell_i \rangle)}$. Sinon, on ne fait rien ;

Étape 2. pour chaque i entre 1 et n, si h_i n'est pas la fonction identité, on copie la valeur du i-ème commutateur dans la i-ème coordonnée en exécutant $f^{(i)}$. Sinon, on ne fait rien.

On peut voir qu'après avoir exécuté ces $2n$ instructions (ou moins si $\Omega(h) \neq n$), on a bien simulé h. Le réseau f de taille sn est bien n-complet en temps $2n$. □

Dans le chapitre précédent, on a vu que l'on pouvait construire des réseaux très difficiles à *séquentialiser*. On a aussi prouvé que ces réseaux difficiles à séquentialiser ont une complexité procédurale importante (avec mémoire arbitrairement grande). Pour tout $f \in \mathrm{F}(m,q)$ et $h \in \mathrm{F}(n,q)$ avec $n \leq m$, on a

$$t_f(h) \geq \mathcal{L}(h \mid m-n) \geq \mathcal{L}(h \mid *) = \boldsymbol{\kappa}(h) + \Omega(h).$$

De ce fait, le Corolaire 9 du chapitre précédent nous donne le résultat énoncé dans le corolaire suivant.

Corolaire 11. *Pour tout m, n, q, le temps de simulation de tout réseau $f \in \mathrm{F}(m,q)$ n-complet est d'au moins $n + \lfloor n/3 \rfloor$. De plus, si $q \geq 4$, alors son temps de simulation est d'au moins $n + \lfloor n/2 - \log_q(n) \rfloor$.*

Cette borne inférieure sur le temps de simulation général est basée sur le fait suivant : il existe un réseau $h \in \mathrm{F}(n,q)$ tel que tout réseau $f \in \mathrm{F}(m,q)$ nécessite un temps d'au moins $\boldsymbol{\kappa}_{n,q}+n$ pour simuler h (même s'il est optimisé pour cela). Toutefois, il est probable qu'un réseau f n-complet ne puisse pas simuler le plus rapidement possible tous les réseaux $h \in \mathrm{F}(n,q)$ à la fois. De ceci découle la conjecture suivante.

Conjecture 4. *Pour tout $f \in \mathrm{F}(n,q)$, n-complet, on a $t_f(n) \geq 2n$.*

6.4 Réseau n-complet de taille minimum

Dans cette section, on va étudier des réseaux n-complets de taille minimum. Pour tout $y, z \in \mathbb{A}^n$, on note $(y \leftrightarrow z)$ la *transposition* de y et z. En d'autres termes, c'est le réseau dans $\mathrm{F}(n,q)$ tel que, pour tout $x \in \mathbb{A}^n$,

$$(y \leftrightarrow z)(x) = \begin{cases} y & \text{si } x = z, \\ z & \text{si } x = y, \\ x & \text{sinon.} \end{cases}$$

On note également $(y \to z)$ l'*affectation* de y à z. En d'autres termes, $(y \to z)$ est la fonction de $\mathrm{F}(n,q)$ telle que

$$(y \to z)(x) = \begin{cases} z & \text{si } x = y, \\ x & \text{sinon.} \end{cases}$$

Enfin, pour $\{x^{(0)}, \ldots, x^{(q^n-1)}\} = \mathbb{A}^n$, on note $c = (x^{(0)} \to x^{(1)} \to \cdots \to x^{(q^n-1)} \to)$ la *permutation circulaire (totale)* de $\mathbb{A}^n$. En d'autres termes, c est une fonction de l'ensemble $\mathrm{F}(n,q)$ telle que, pour tout $i \in [0, q^n[$, $c(x^{(i)}) = x^{(i+1)}$ (l'addition étant faite modulo q^n). En termes de graphe de transition, c n'a qu'un seul cycle limite (qui est donc de taille q^n).

La proposition 11, nous donne un ensemble de trois fonctions dans $\mathrm{F}(n,q)$ (une permutation circulaire, une transposition et une affectation) qui génèrent $\mathrm{F}(n,q)$. Ceci veut dire que si le réseau $f \in \mathrm{F}(m,q)$ simule ces trois fonctions, alors il est n-complet. Ce résultat a été prouvé dans [119] et repris notamment dans [33].

Proposition 11. *Considérons une permutation circulaire c sur $\mathbb{A}^n$, une transposition $k = (y \leftrightarrow y')$ telle que $c(y) = y'$ et une affectation $d = (z \to z')$. Ces trois fonctions génèrent $\mathrm{F}(n,q)$.*

Il a été démontré dans [39] qu'à part dans le cas $(n,q) = (2,2)$, il existe un réseau $f \in \mathrm{F}(n,q)$ qui simule l'ensemble des réseaux $h \in \mathrm{F}(n,q)$ bijectifs. Dans le théorème suivant, on montre qu'il n'existe aucun réseau n-complet de taille n mais que l'on peut en construire un de taille $n+1$.

Théorème 25. *Pour tout $n \geq 1$ et pour toute taille d'alphabet $q \geq 2$, le plus petit réseau n-complet est de taille $n+1$.*

Démonstration. Montrons d'abord qu'il n'existe pas de réseau n-complet de taille n.

Affirmation 12. *Aucun réseau $f \in \mathrm{F}(n,q)$ ne peut simuler à la fois un réseau bijectif $h \in \mathrm{F}(n,q)$ sans fonction locale triviale et un réseau non bijectif $g \in \mathrm{F}(n,q)$.*

Démonstration. Considérons donc un réseau bijectif $h \in \mathrm{F}(n,q)$ sans aucune fonction locale triviale et un réseau non bijectif $g \in \mathrm{F}(n,q)$, tous les deux simulés par un réseau $f \in \mathrm{F}(n,q)$ (par exemple, on peut avoir $h : x \mapsto (x_1+1, x_2+1, \dots, x_n+1)$ et $g : x \mapsto (\mathtt{0})^n$). Si toutes les instructions $f^{(i)}$ avec $i \in [n]$ étaient bijective alors, pour tout $w \in [n]^*$, la fonction f^w serait bijective et f ne simulerait pas g. En effet, la fonction obtenue en associant des instructions bijectives est elle-même bijective. Il doit donc exister une coordonnée $i \in [n]$ telle que l'instruction $f^{(i)}$ est singulière (*i.e.* non bijective).

Prenons un mot $w \in [n]^*$ tel que $f^w = h$. Clairement, i doit apparaitre dans w et donc le réseau $h = f^w$ n'est pas bijectif, ce qui contredit sa définition.

■

On va maintenant montrer que, pour tout $n \geq 1$ et $q \geq 2$, il existe un réseau n-complet de taille $n+1$. On s'occupe d'abord du cas $n = 1$.

Affirmation 13. *Pour tout $q \geq 2$, il existe un réseau $f \in \mathrm{F}(2,q)$ qui est 1-complet.*

Démonstration. Définissons $f \in \mathrm{F}(2,q)$ comme suit. Pour tout $x \in \mathbb{A}^n$,

$$f_1(x) = \begin{cases} x_1 + 1 & \text{si } x_1 = x_2, \\ \mathtt{1} & \text{si } x_1 = \mathtt{0} \text{ et } x_2 = q-1, \\ \mathtt{0} & \text{si } x_1 = \mathtt{1} \text{ et } x_2 = \mathtt{0}, \\ x_1 & \text{sinon,} \end{cases}$$
$$f_2(x) = x_1.$$

Montrons que f est 1-complet. Premièrement, on peut voir que $f^{(2,1)}$ simule la permutation circulaire $c = (\mathtt{0} \to \mathtt{1} \to \cdots \to q-1 \to)$. En effet,

$$(x_1, x_2) \xrightarrow{f^{(2)}} (x_1, x_1) \xrightarrow{f^{(1)}} (x_1+1, x_1).$$

Deuxièmement, $f^{((2,1)^q,1)}$ simule la transposition $k = (\mathtt{0} \leftrightarrow \mathtt{1})$. En effet,

$$\begin{aligned} & (\mathtt{0}, x_2) \xrightarrow{f^{(2,1)^q}} (\mathtt{0}, q-1) \xrightarrow{f^{(1)}} (\mathtt{1}, q-1), \\ & (\mathtt{1}, x_2) \xrightarrow{f^{(2,1)^q}} (\mathtt{1}, \mathtt{0}) \xrightarrow{f^{(1)}} (\mathtt{0}, \mathtt{0}), \\ \forall i \in \mathbb{A} \setminus \{\mathtt{0}, \mathtt{1}\}, \quad & (i, x_2) \xrightarrow{f^{(2,1)^q}} (i, i-1) \xrightarrow{f^{(1)}} (i, i-1). \end{aligned}$$

Troisièmement, $f^{((2,1)^q,1,1)}$ simule l'affectation $d = (\mathtt{0} \to \mathtt{1})$. En effet,

$$\begin{array}{rlllll} & (0,x_2) & \xrightarrow{f^{(2,1)^q}} (0,q-1) & \xrightarrow{f^{(1)}} (1,q-1) & \xrightarrow{f^{(1)}} (1,q-1), \\ & (1,x_2) & \xrightarrow{f^{(2,1)^q}} (1,0) & \xrightarrow{f^{(1)}} (0,0) & \xrightarrow{f^{(1)}} (1,0), \\ \forall i \in \mathbb{A} \setminus \{0,1\}, & (i,x_2) & \xrightarrow{f^{(2,1)^q}} (i,i-1) & \xrightarrow{f^{(1)}} (i,i-1) & \xrightarrow{f^{(1)}} (i,i-1). \end{array}$$

■

On s'intéresse ensuite aux cas où $n \geq 2$.

Affirmation 14. *Pour tout $n \geq 2$ et $q \geq 2$, il existe un réseau $f \in \mathrm{F}(n+1,q)$ qui est n-complet.*

Démonstration. Dans cette preuve, on va représenter chaque nombre $t \in [0, q^n[$ par une configuration $y^{(t)} \in \mathbb{A}^n$. Plus précisément, pour tout $t = t_1 q^{n-1} + t_2 q^{n-2} + \cdots + t_n$ avec $t_1, t_2, \ldots, t_n \in \mathbb{A}$, on va prendre $y^{(t)} = (t_1, t_2, \ldots, t_n)$. En particulier, $y^{(0)} = (\mathtt{0})^n$, $y^{(1)} = (\mathtt{0})^{n-1}\mathtt{1}$ et $y^{(q)} = (\mathtt{0})^{n-2}\mathtt{10}$. Définissons $f \in \mathrm{F}(n+1,q)$ comme suit : pour tout $x \in \mathbb{A}^n$,

$$f_{n+1}(x) = x_n,$$

$$f_n(x) = \begin{cases} x_n + 1 & \text{si } x_n = x_{n+1}, \\ \mathtt{1} & \text{sinon si } x_{[n]} = y^{(0)}, \\ \mathtt{0} & \text{sinon si } x_{[n]} = y^{(1)}, \\ x_n & \text{sinon,} \end{cases}$$

$$f_{n-1}(x) = \begin{cases} x_{n-1} + 1 & \text{si } x_n = \mathtt{0} \text{ et } x_{n+1} \neq x_n, \\ \mathtt{1} & \text{si } x_n = x_{n+1} \text{ et } x_{[n]} = y^{(0)}, \\ x_{n-1} & \text{sinon,} \end{cases}$$

$$\forall i \in [n-2],\ f_i(x) = \begin{cases} x_i + 1 & \text{si } x_{[i,n]} = (\mathtt{0})^{n-i+1}, \\ x_i & \text{sinon.} \end{cases}$$

Premièrement, on peut voir que $f^{(n+1,n,n-1,\ldots,1)}$ simule la permutation circulaire $c \in \mathrm{F}(n,q)$ telle que, pour tout $t \in [0, q^n[$, $c(y^{(t)}) = y^{(t+1)}$, $t+1$ étant calculé modulo q^n. En effet, pour tout $t \in [0, q^n[$ et tout $x_{n+1} \in \mathbb{A}$,

$$y^{(t)} x_{n+1} \xrightarrow{f^{(n+1)}} y^{(t)} y_n^{(t)} \xrightarrow{f^{(n,n-1,\ldots,1)}} y^{(t+1)} y_n^{(t)}.$$

Deuxièmement, $f^{((n+1,n)^q,n)}$ simule la transposition $k = (y^{(0)} \leftrightarrow y^{(1)})$, car

$$\begin{array}{lll} \forall t \in [2, q^n[,\ y^{(t)}x_{n+1} & \xrightarrow{f^{((n+1,n)^q)}} y^{(t)}(y_n^{(t)} - 1) & \xrightarrow{f^{(n)}} y^{(t)}(y_n^{(t)} - 1), \\ y^{(0)}x_{n+1} & \xrightarrow{f^{((n+1,n)^q)}} y^{(0)}(q-1) & \xrightarrow{f^{(n)}} y^{(1)}(q-1), \\ y^{(1)}x_{n+1} & \xrightarrow{f^{((n+1,n)^q)}} y^{(1)}0 & \xrightarrow{f^{(n)}} y^{(0)}0. \end{array}$$

Enfin, $f^{(n+1,n-1)}$ simule l'affectation $d = (y^{(0)} \to y^{(q)})$. En effet,

$$\begin{array}{lll} \forall t \in [1, q^n[,\ y^{(t)}x_{n+1} & \xrightarrow{f^{(n+1)}} y^{(t)}y_n^{(t)} & \xrightarrow{f^{(n-1)}} y^{(t)}y_n^{(t)}, \\ y^{(0)}x_{n+1} & \xrightarrow{f^{(n+1)}} y^{(0)}0 & \xrightarrow{f^{(n-1)}} y^{(q)}0. \end{array}$$

■

Les affirmations 12, 13 et 14 précédentes démontrent directement le théorème.

□

On remarque que le temps de simulation d'un réseau n-complet de taille $n+1$ est nécessairement très grand. En effet, on a besoin de $s = (q^n)^{(q^n)}$ mots $w^{(1)}, \ldots, w^{(s)}$ pour simuler tous les réseaux $h \in \mathrm{F}(n, q)$ à partir du réseau n-complet $f \in \mathrm{F}(n+1, q)$. De plus, ces mots sont seulement sur l'alphabet $[n+1]$. On remarque qu'avec uniquement des mots de sur l'alphabet $[n+1]$ de taille t, le réseau f peut simuler au mieux $1 + (n+1) + (n+1)^2 + \cdots + (n+1)^t = (n+1)^{t+1} - 1$ réseaux $h \in \mathrm{F}(n, q)$. De ce fait, le plus long des mots $w^{(i)}$ est de taille au moins $\lceil \log_{n+1}(s+1) - 1 \rceil$.

6.5 Réseau n, q-complet

Dans la section 6.4, on a vu qu'il n'existait pas de réseau $f \in \mathrm{F}(n, q)$ qui simule l'ensemble des réseaux $h \in \mathrm{F}(n, q)$. On a alors considéré un réseau de taille $n+1$ qui peut les simuler tous. Dans cette section, à la place d'augmenter la taille du réseau considéré, on va choisir d'augmenter la taille de l'alphabet. On va alors dire qu'un réseau $f \in \mathrm{F}(n, q')$ est n, q-complet s'il peut simuler tous les réseaux $h \in \mathrm{F}(n, q)$.

Ici, nous nous focalisons sur deux définitions alternatives de simulation avec un alphabet plus grand (et donc de n,q-complétude) qui nous paraissent pertinentes.

Définition 15 (n,q-complétude par facteur)**.** *Soit $\mathbb{A}$ et $\mathbb{B}$ deux alphabets de cardinalités respectives q et q'. On dit que $f \in \mathrm{F}(m, q')$ avec $m \geq n$ est* n,q-complet par facteur *s'il existe une fonction $\mu : \mathbb{B} \to \mathbb{A}$ surjective telle que, pour tout $h \in \mathrm{F}(n, q)$, il existe un mot $w \in [n]^*$ tel que*

$$\varphi \circ f^w = h \circ \varphi \text{ avec } \varphi : x \mapsto \mu(x_1)\mu(x_2)\ldots\mu(x_n).$$

Remarque 6. *On pourrait ajouter comme condition à cette définition que la fonction μ soit* équilibrée *:*

$$\forall b, b' \in \mathbb{B}, |\mu^{-1}(b)| = |\mu^{-1}(b')|.$$

Avec cette définition, les résultats énoncés dans le théorème 26 deviennent optimaux du point de vue de la taille de l'alphabet.

Définition 16 (n,q-complet avec initialisation)**.** *Soit $\mathbb{A}$ et $\mathbb{B}$ deux alphabets de cardinalités respectives q et q'. On dit que $f \in \mathrm{F}(m, q')$ avec $m \geq n$ est* n,q-complet avec initialisation *si pour tout réseau $h \in \mathrm{F}(n, q)$, il existe un mot $w \in [n]^*$ tel que*

$$\forall z \in \mathbb{A}^m, f^w_{[n]}(z) = h(z_{[n]}).$$

Remarque 7. *On peut voir que, pour ces deux définitions, l'alphabet supplémentaire peut nous permettre de créer des réseaux n, q-complets qui ont un temps de simulation inférieur à $2n$.*

En effet, en reprenant l'idée des commutateurs du théorème 24, on peut encoder l'image de $h(x)$. Ceci va prendre un temps $\log_{q'}(q^n)$, avec q' la taille du plus grand alphabet. Ensuite, il faut recopier l'image, ce qui prend un temps n. On peut donc créer un réseau $f \in \mathrm{F}(m, q')$ n,q-complet par facteur en temps $n + \log_{q'}(q^n)$.

Avec la deuxième définition, on pourrait adopter une stratégie différente. On peut indiquer en une seule étape de temps la fonction h que l'on veut simuler grâce à q^n instructions $f^{(i)} : x_i \leftarrow \alpha$ (avec $\alpha > q$). Après cela, on pourrait calculer la fonction h en un temps $\log_{q'}(q^{\kappa(h)}) + \Omega(h)$ en utilisant la technique vu au chapitre précédent (voir la démonstration du lemme 21).

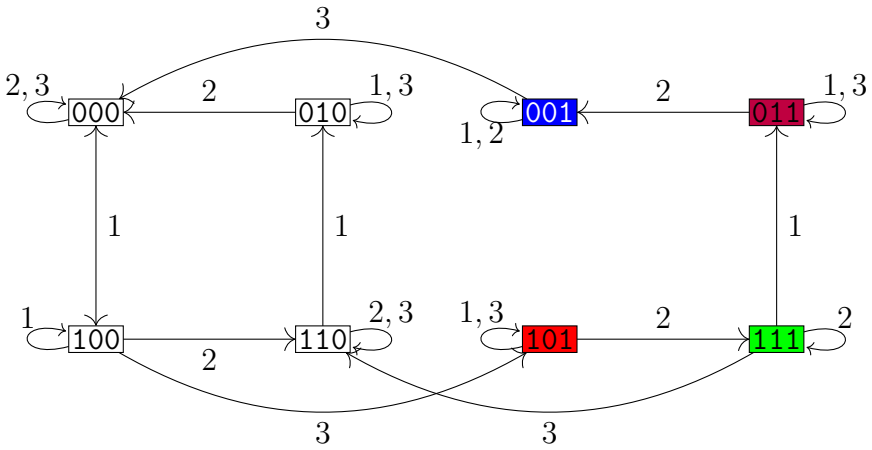


FIGURE 6.1 – Illustration du comportement asynchrone (*i.e.* séquentiel non déterministe) de la fonction $\Pi : \{0,1\}^3 \to \{0,1\}^3$ dans le théorème 26.

Le théorème 26 ci-dessous montre que, pour $n \geq 3$, on peut construire un réseau de taille n avec un alphabet de taille $2q$ qui est n,q-complet par facteur. Pour donner

une intuition de la preuve, on va voir notre alphabet $[0, 2q[$ comme la concaténation de l'alphabet $\mathbb{A}$ et d'un alphabet booléen $\{0, 1\}$. Grâce à cet alphabet $\{0, 1\}$, on va pouvoir indiquer quelle opération doit être effectuée sur l'alphabet $\mathbb{A}$. On a deux problèmes. Le premier est que l'on ne connait pas la valeur de notre configuration initiale sur l'alphabet $\{0, 1\}$. Ce problème peut être résolu grâce à l'automate synchronisant représenté dans la figure 6.1 : en effectuant une séquence de mise à jour, on va se placer dans l'état 101 quel que soit notre état de départ. Deuxièmement, on doit faire attention, avec cette séquence de réglage, de ne pas altérer la configuration sur l'alphabet $\mathbb{A}$.

Théorème 26. *Pour tout $q \geq 2$ et $n \geq 3$, il existe $f \in \mathrm{F}(n, 2q)$ qui est n,q-complet par facteur.*

Démonstration. Posons $\mathbb{A} = [0, q[$, $q' = 2q$ et $\mathbb{B} = [0, q'[$. Considérons $\mu : \mathbb{B} \to \mathbb{A}$ et $\lambda : \mathbb{B} \to \{0, 1\}$ tels que, pour tout $b = \ell q + a \in \mathbb{B}$ (ℓ et a sont respectivement le quotient et le reste de la division de b par q), on a $\mu(b) = a$ et $\lambda(b) = \ell$. Définissons les deux fonctions ϕ et φ telles que

$$\forall z \in \mathbb{B}^n, \phi(z) = (\lambda(z_1), \lambda(z_2), \lambda(z_3)) \text{ et } \varphi(z) = (\mu(z_1), \mu(z_2), \dots, \mu(z_n)).$$

Par abus de notation et pour simplifier la rédaction, dans la suite de cette preuve, on écrira une configuration $z \in \mathbb{B}^n$ comme le vecteur xy avec $x = \varphi(z)$ et $y = \phi(z)$. On peut faire ceci, car on n'utilisera pas les valeurs $\lambda(z_4), \lambda(z_5)$...

Maintenant, on va considérer la fonction $\Pi : \{0, 1\}^3 \to \{0, 1\}^3$ représentée dans la figure 6.1 et la fonction $\Psi : \mathbb{B}^n \to \mathbb{A}^n$ définie ci-dessous. Pour tout $z = xy \in \mathbb{B}^n$,

$$\Psi_1(z) = \begin{cases} x_1 + 1 & \text{si } y = 101, \\ 1 & \text{si } x = (0)^n \text{ et } (y = 011 \text{ ou } y = 001), \\ 0 & \text{si } x = 1(0)^{n-1} \text{ et } y = 011, \\ x_1 & \text{sinon,} \end{cases}$$

$$\Psi_2(z) = \begin{cases} x_2 + 1 & \text{si } x_1 = 0 \text{ et } y = 111, \\ x_2 & \text{sinon,} \end{cases}$$

$$\Psi_3(z) = \begin{cases} x_3 + 1 & \text{si } x_1 = x_2 = 0 \text{ et } y = 011, \\ x_3 & \text{sinon,} \end{cases}$$

$$\forall i \in [4, n],\ \Psi_i(z) = \begin{cases} x_i + 1 & \text{si } x_1 = x_2 = \cdots = x_{i-1} = 0, \\ x_i & \text{sinon.} \end{cases}$$

Premièrement, définissons le réseau $f \in \mathrm{F}(n, q')$ tel que,

— $\forall i \in [1, 3], f^{(i)} : z = xy \mapsto \Psi^{(i)}(z)\Pi^{(i)}(y)$;

— $\forall i \in [4, n]$, $f^{(i)} : z = xy \mapsto \Psi^{(i)}(z)y$.

Prouvons maintenant que f est n,q-complet par facteur.

Premièrement, prouvons que, pour tout $z = xy \in \mathbb{B}^n$, on a bien $f^{((3)^q,2,3,1,2,1,3)}(xy) = x101$. Remarque d'abord que, quand on met à jour la troisième coordonnée q fois de suite, il y a deux cas :

— on a $y \neq 011$ ou $x_1 \neq 0$ ou $x_2 \neq 0$ et la valeur de x n'est pas modifiée ;

— on a $y = 011$ et $x_1 = x_2 = 0$, et on applique alors q fois l'instruction $x_3 \leftarrow x_3 + 1$. De ce fait après les q mises à jour, on obtient $\varphi(f^{((3)^q)}(z)) = (x_1, x_2, x_3 + q, \dots) = (x_1, x_2, x_3, \dots) = x$.

Pour vérifier que la séquence de mises à jour $((3)^q, 2, 3, 1, 1, 2, 1, 3)$ ne modifie pas la valeur de x, on a seulement besoin de vérifier les trois hypothèses suivantes :

— on ne met pas à jour la coordonnée 1 quand $y \in \{101, 011, 001\}$;

— on ne met pas à jour la coordonnée 2 quand $y = 111$;

— quand on met à jour la coordonnée 3 et que $y = 011$, on la met à jour q fois de suite.

Par simple application de $f^{((3)^q,2,3,1,1,2,1,3)}$, on obtient

$$
\begin{array}{llllllll}
x000 \xrightarrow{(3)^q} x000 \xrightarrow{2} x000 \xrightarrow{3} x000 \xrightarrow{1} x100 \xrightarrow{1} x000 \xrightarrow{2} x000 \xrightarrow{1} x100 \xrightarrow{3} x101, \\
x100 \xrightarrow{(3)^q} x101 \xrightarrow{2} x111 \xrightarrow{3} x110 \xrightarrow{1} x010 \xrightarrow{1} x010 \xrightarrow{2} x000 \xrightarrow{1} x100 \xrightarrow{3} x101, \\
x010 \xrightarrow{(3)^q} x010 \xrightarrow{2} x000 \xrightarrow{3} x000 \xrightarrow{1} x100 \xrightarrow{1} x000 \xrightarrow{2} x000 \xrightarrow{1} x100 \xrightarrow{3} x101, \\
x110 \xrightarrow{(3)^q} x110 \xrightarrow{2} x110 \xrightarrow{3} x110 \xrightarrow{1} x010 \xrightarrow{1} x010 \xrightarrow{2} x000 \xrightarrow{1} x100 \xrightarrow{3} x101, \\
x001 \xrightarrow{(3)^q} x000 \xrightarrow{2} x000 \xrightarrow{3} x000 \xrightarrow{1} x100 \xrightarrow{1} x000 \xrightarrow{2} x000 \xrightarrow{1} x100 \xrightarrow{3} x101, \\
x011 \xrightarrow{(3)^q} x011 \xrightarrow{2} x001 \xrightarrow{3} x000 \xrightarrow{1} x100 \xrightarrow{1} x000 \xrightarrow{2} x000 \xrightarrow{1} x100 \xrightarrow{3} x101, \\
x101 \xrightarrow{(3)^q} x101 \xrightarrow{2} x111 \xrightarrow{3} x110 \xrightarrow{1} x010 \xrightarrow{1} x010 \xrightarrow{2} x000 \xrightarrow{1} x100 \xrightarrow{3} x101, \\
x111 \xrightarrow{(3)^q} x110 \xrightarrow{2} x110 \xrightarrow{3} x110 \xrightarrow{1} x010 \xrightarrow{1} x010 \xrightarrow{2} x000 \xrightarrow{1} x100 \xrightarrow{3} x101.
\end{array}
$$

On remarque que cette séquence de mises à jour nous permet de passer d'une configuration xy avec y quelconque à la configuration $x101$. Deuxièmement, montrons qu'en partant de $z = x101 \in \mathbb{B}^n$, on peut obtenir $c(x)011$ avec c la permutation circulaire

$$c : ((0)^n \to 1(0)^n \to \cdots \to (q-1)(0)^{n-1} \to 01(0)^{n-2} \to \dots).$$

En effet, en appliquant $f^{(1,2,2,1,(3,4,\dots,n))}$ on obtient

$$
\begin{array}{l}
x101 \xrightarrow{1} c^{[1]}(x)101 \xrightarrow{2} c^{[1]}(x)111 \xrightarrow{2} c^{[2]}(x)111 \xrightarrow{1} c^{[2]}(x)011 \xrightarrow{3} c^{[3]}(x)011 \\
\xrightarrow{4} c^{[4]}(x)011 \xrightarrow{5} \dots \xrightarrow{n} c(x)011.
\end{array}
$$

Maintenant, observons que, pour tout $z = x101 \in \mathbb{B}^n$, on peut calculer la transposition

$k = ((0)^n \leftrightarrow 1(0)^{n-1})$ avec $f^{(2,1,1)}$. En effet,

$$x101 \xrightarrow{2} x111 \xrightarrow{1} x011 \xrightarrow{1} k(x)011.$$

De la même façon, on peut voir que, pour tout $z = x101 \in \mathbb{B}^n$, on peut simuler l'affectation $d = ((0)^n \to 1(0)^{n-1})$ en exécutant $f^{(2,1,2,1)}$. En effet,

$$x101 \xrightarrow{2} x111 \xrightarrow{1} x011 \xrightarrow{2} x001 \xrightarrow{1} d(x)001.$$

Les fonctions c, k et d génèrent l'ensemble $\mathrm{F}(n, q)$. De ce fait, f est n,q-complet par facteur.

□

Remarque 8. *Cette construction n'utilise en fait l'alphabet plus grand que sur* 3 *coordonnées.*

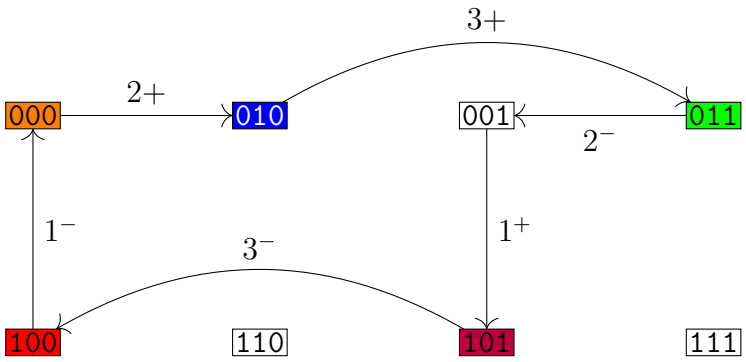


FIGURE 6.2 – Illustration des fonctions ⊞ et ⊟ dans le théorème 27

.

Le théorème 27 ci-dessous montre que, pour n suffisamment grand, un alphabet avec une unique lettre supplémentaire suffit pour construire un réseau n,q-complet avec initialisation.

Théorème 27. *Pour tout $q \geq 2$ et $n \geq 3q$, il existe $f \in \mathrm{F}(n, q+1)$ qui soit n,q-complet avec initialisation.*

Démonstration. Posons $\mathbb{A} = [0, q[$, $\mathbb{B} = [0, q'[$ avec $q' = q+1$ et $\{\alpha\} = \mathbb{B} \setminus \mathbb{A}$. L'idée principale de cette preuve est d'encoder dans chaque configuration $z \in \mathbb{B}^q$ à la fois une configuration $x \in \mathbb{A}^q$ et une valeur booléenne $y \in \{0, 1\}$. Notons Z^+ l'ensemble des configurations $z \in \mathbb{B}^q$ telles qu'il existe une *unique* coordonnée $j \in [q]$ telle que $z_j = \alpha$. Notons également Z^- l'ensemble des configurations $z \in \mathbb{B}^q$ telles qu'il n'existe aucune telle coordonnée j et posons $Z = Z^+ \cup Z^-$.

Définissons $\lambda : \mathbb{B}^q \to \{\mathtt{0}, \mathtt{1}\}$, la fonction telle que $\lambda(z)$ nous donne la valeur booléenne encodée dans z. Pour tout $z \in \mathbb{B}^q$, posons $\lambda(z) = \begin{cases} \mathtt{0} & \text{si } z \in Z^- \\ \mathtt{1} & \text{sinon} \end{cases}$. De plus, définissons la fonction $\mu : \mathbb{B}^q \to \mathbb{A}^q$ telle que $\mu(z)$ nous donne la configuration $\mu(z) \in \mathbb{A}^q$ encodée dans z :

— si $z \notin Z$ alors peu importe la valeur de $\mu(z)$ (on peut par exemple fixer $\mu(z) = (\mathtt{0})^q$) ;

— si $z \in Z^-$ alors $\mu(z) = z$;

— enfin, si $z \in Z^+$ alors on a une unique coordonnée $j \in [q]$ telle que $z_j = \alpha$ et on pose

$$(\mu(z))_{[q]\setminus\{j\}} = z_{[q]\setminus\{j\}} \text{ et } (\mu(z))_j = (j-1) - \sum_{i \in [q]\setminus\{j\}} z_j.$$

Illustrons ces deux définitions avec un exemple. Imaginons que l'on ait $q = 4$ et que l'on veuille encoder une valeur booléenne $y \in \{\mathtt{0}, \mathtt{1}\}$ et la configuration $x = (x_1, x_2, x_3, x_4) \in \mathbb{A}^q$ avec $\sum_{i \in [q]} x_i = 2$ dans une configuration $z \in Z$. Alors, la configuration $z \in Z$ va être définie comme suit en fonction la valeur de y :

x	(x_1, x_2, x_3, x_4)	
y	$\mathtt{0}$	$\mathtt{1}$
z	(x_1, x_2, x_3, x_4)	(x_1, x_2, α, x_4)

.

Définissons les deux fonctions $\boxplus : Z \mapsto Z$ et $\boxminus : Z \mapsto Z$ telles que $\forall z \in Z$,

$$\forall j \in [q],\ \boxplus_j(z) = \begin{cases} \alpha & \text{si } z \in Z^- \text{ et } j - 1 = \sum_{i \in [q]} z_i, \\ z_j & \text{sinon,} \end{cases}$$

$$\forall j \in [q],\ \boxminus_j(z) = \begin{cases} (j-1) - \sum_{i \in [q]\setminus\{j\}} z_i & \text{si } z_i = \alpha, \\ z_j & \text{sinon.} \end{cases}$$

On remarque qu'en appliquant $\boxplus^{(1,2,\ldots,q)}$ sur une configuration $z \in Z$, on obtient la configuration $z' \in Z^+$ telle que $\mu(z) = \mu(z')$. De même, en appliquant $\boxminus^{(1,2,\ldots,q)}$ sur z, on obtient la configuration $z' \in Z^-$ telle que $\mu(z) = \mu(z')$. Pour tout $i \in [3]$ et $j \in [q]$, notons $\Phi^i_j = (i-1)q + j$. Pour tout $i \in [3]$, notons $\Phi^i = \{\Phi^i_1, \ldots \Phi^i_q\}$. Enfin, notons $R = [3q+1, n]$. On a donc $[n] = \Phi^1 \cup \Phi^2 \cup \Phi^3 \cup R$. Définissons la fonction $\varphi : \mathbb{B}^n \to \mathbb{A}^n$ comme suit :

$$\varphi : z \mapsto \mu(z_{\Phi_1})\mu(z_{\Phi_2})\mu(z_{\Phi_3})z_R \text{ si } z_R \in \mathbb{A}^{|R|},$$

et peu importe sa valeur sinon. De plus, définissons la fonction $\phi : \mathbb{B}^n \to \{\mathtt{0}, \mathtt{1}\}^3$ comme suit :

$$\phi : z \mapsto \lambda(z_{\Phi^1})\lambda(z_{\Phi^2})\lambda(z_{\Phi^3}).$$

Par abus de langage et pour simplifier la notation, représentons toute configuration $z \in \mathbb{B}^n$ comme un couple $xy = \varphi(z)\phi(z)$.

Définissons maintenant la fonction $\Psi : \mathbb{B}^n \to \mathbb{A}^n$ qui va indiquer comment notre fonction f doit modifier la configuration x. Pour tout $z = xy \in \mathbb{B}^n$, pour tout $i \in [n]$,

$$\text{si } i = \Phi_1^1,\ \Psi_i(z) = \begin{cases} x_i + 1 & \text{si } y = 010, \\ 1 & \text{si } y = 011 \text{ et } x = (0)^n, \\ 0 & \text{si } y = 011 \text{ et } x = 1(0)^{n-1}, \\ x_i & \text{sinon,} \end{cases}$$

$$\text{sinon si } i = \Phi_1^2,\ \Psi_i(z) = \begin{cases} x_i + 1 & \text{si } y = 001 \text{ et } x_1 = x_2 = \cdots = x_{i-1} = 0, \\ 1 & \text{si } y = 101 \text{ et } x = (0)^n, \\ x_i & \text{sinon,} \end{cases}$$

$$\text{sinon si } i = \Phi_1^3,\ \Psi_i(z) = \begin{cases} x_i + 1 & \text{si } y = 000 \text{ et } x_1 = x_2 = \cdots = x_{i-1} = 0, \\ x_i & \text{sinon,} \end{cases}$$

$$\text{sinon, } \Psi_i(z) = \begin{cases} x_i + 1 & \text{si } x_1 = x_2 = \cdots = x_{i-1} = 0, \\ x_i & \text{sinon.} \end{cases}$$

Enfin, définissons notre réseau $f \in \mathrm{F}(n, q')$ comme suit :

— pour tout arc $y \xrightarrow{i^+} y'$ dans la figure 6.2 et, pour toute configuration $z \in \mathbb{B}^n$ telle que $\phi(z) = y$, on pose $f_{\Phi^i}(z) = \boxplus(z_{\Phi^i})$ si $z_{\Phi^i} \in Z$ et sinon peu importe ;

— pour tout arc $y \xrightarrow{i^-} y'$ dans la figure 6.2 et, pour toute configuration $z \in \mathbb{B}^n$ telle que $\phi(z) = y$, on pose $f_{\Phi^i}(z) = \boxminus(z_{\Phi^i})$ si $z_{\Phi^i} \in Z$ et sinon peu importe ;

— pour tout $j \in [n]$ et toute configuration $z \in \mathbb{B}^n$ tels que $f_j(z)$ n'a pas encore été définie, on pose juste $f_j(z) = \Psi_j(z)$.

Pour tout $i \in [3]$, définissons la séquence de mise à jour $\tilde{\imath} = (\Phi_1^i, \Phi_2^i, \ldots, \Phi_q^i)$ et $\tilde{4} = (3q + 1, \ldots, n)$.

Premièrement, montrons que, pour tout $x \in \mathbb{A}^n$, la fonction $f^{(\tilde{2},\tilde{1},\tilde{3},\tilde{2},\tilde{1},\tilde{3},\tilde{2},\tilde{1},\tilde{3},\tilde{4})}$ associe $x000$ à $c(x)000$ avec c la permutation circulaire décrite dans la preuve du théorème 26 :

$$x000 \xrightarrow{\tilde{2}} x010 \xrightarrow{\tilde{1}} c^{[q]}(x)010 \xrightarrow{\tilde{3}} c^{[q]}(x)011 \xrightarrow{\tilde{2}} c^{[q]}(x)001 \xrightarrow{\tilde{1}} c^{[q]}(x)101$$
$$\xrightarrow{\tilde{3}} c^{[q]}(x)100 \xrightarrow{\tilde{2}} c^{[2q]}(x)100 \xrightarrow{\tilde{1}} c^{[2q]}(x)000 \xrightarrow{\tilde{3}} c^{[3q]}(x)000 \xrightarrow{\tilde{4}} c(x)000.$$

Deuxièmement, montrons que, pour tout $x \in \mathbb{A}^n$, la fonction $f^{(\tilde{2},\tilde{3},\tilde{1},\tilde{2},\tilde{1},\tilde{3},\tilde{1})}$ associe $x000$ à

$k(x)000$ avec k la transposition $((0)^n \leftrightarrow 1(0)^{n-1})$:

$$x000 \xrightarrow{\tilde{2}} x010 \xrightarrow{\tilde{3}} x011 \xrightarrow{\tilde{1}} k(x)011 \xrightarrow{\tilde{2}} k(x)001 \xrightarrow{\tilde{1}} k(x)101 \xrightarrow{\tilde{3}} k(x)100 \xrightarrow{\tilde{1}} k(x)000.$$

Pour finir, montrons que, pour tout $x \in \mathbb{A}^n$, la fonction $f^{(\tilde{2},\tilde{3},\tilde{2},\tilde{1},\tilde{2},\tilde{3},\tilde{1})}$ associe $x000$ à $d(x)000$ avec d l'affectation $((0)^n \to (0)^q 1(0)^{n-q-1})$.

$$(x000) \xrightarrow{\tilde{2}} x010 \xrightarrow{\tilde{3}} x011 \xrightarrow{\tilde{2}} x001 \xrightarrow{\tilde{1}} x101 \xrightarrow{\tilde{2}} x101 \xrightarrow{\tilde{3}} x100 \xrightarrow{\tilde{1}} d(x)000.$$

Les trois fonctions c, k et d génèrent toutes les fonctions de $\mathrm{F}(n,q)$. De ce fait, f est n,q-complet avec initialisation.

□

6.6 Simuler un réseau sans mémoire

Un réseau $h \in \mathrm{F}(n,q)$ est *simulable sans mémoire* s'il peut être simulé par un réseau $f \in \mathrm{F}(n,q)$ qui est donc de même taille et travaille sur le même alphabet. On peut voir que le cas $n = 1$ n'est pas intéressant : tous les réseaux $h \in \mathrm{F}(1,q)$ se simulent eux-mêmes ($h = h^{(1)}$). On va donc considérer dans la suite que $n \geq 2$.

On a vu plus tôt que, pour $(n,q) \neq (2,2)$, il existe un réseau $f \in \mathrm{F}(n,q)$ qui simule tous les réseaux bijectifs $h \in \mathrm{F}(n,q)$. Par conséquent, tous les réseaux bijectifs pour $n > 2$ ou $q > 2$ sont simulables sans mémoire. On va montrer qu'à l'inverse, il existe des réseaux bijectifs $h \in \mathrm{F}(2,2)$ non simulables sans mémoire.

Proposition 12. *La permutation circulaire* $(00 \to 01 \to 11 \to 10 \to)$ *dans* $\mathrm{F}(2,2)$ *n'est pas simulable sans mémoire.*

Démonstration. Notons h cette permutation circulaire. Supposons par l'absurde qu'il existe un réseau $f \in \mathrm{F}(2,2)$ et un mot $w \in [2]^*$ tels que $f^w = h$. Comme le réseau h n'a pas de fonction locale triviale et est bijectif, les coordonnées 1 et 2 apparaissent dans w et les instructions $f^{(1)}$ et $f^{(2)}$ sont bijectives.

Cela signifie que si $f^{(1)}(x) = x + e^1$ alors $f^{(1)}(x + e^1) = x$ pour tout $x \in \{0,1\}^n$. Il n'y a donc que quatre instuctions possibles pour $f^{(1)}$:

- la fonction identité $x \mapsto x$;
- la simple transposition $(00 \leftrightarrow 10)$;
- la simple transposition $(01 \leftrightarrow 11)$;
- la double transposition $(00 \leftrightarrow 10)(01 \leftrightarrow 11)$.

L'instruction $f^{(1)}$ ne peut pas être l'identité sinon $f_1^w(\mathtt{01}) = \mathtt{0} \neq \mathtt{1} = h_1(\mathtt{01})$. L'instruction $f^{(1)}$ ne peut pas être la double transposition non plus car, sinon, on pourrait prouver l'invariant suivant : pour tout $w \in [n]^*$, $f_1^w(\mathtt{00}) = f_1^w(\mathtt{01})$. Or, $h_1(\mathtt{00}) = \mathtt{0} \neq \mathtt{1} = h_1(\mathtt{01})$. De ce fait, $f^{(1)}$ est une simple transposition et il existe une lettre $a \in \{\mathtt{0}, \mathtt{1}\}$ telle que, pour tout $x \in \{\mathtt{0}, \mathtt{1}\}^2$ avec $x_2 = a$, on a $f^{(1)}(x) = x$.

En reprenant le même raisonnement pour $f^{(2)}$, on prouve qu'il existe une lettre $b \in \{\mathtt{0}, \mathtt{1}\}$ telle que, pour tout $x \in \{\mathtt{0}, \mathtt{1}\}^2$ avec $x_1 = b$, on a $f^{(2)}(x) = x$. En prenant $y = ba$, on obtient $f^{(1)}(y) = y$, $f^{(2)}(y) = y$ et donc $h(y) = f^w(y) = y$, ce qui contredit la définition de h.

Il n'existe donc aucun réseau $f \in \mathrm{F}(2, 2)$ qui simule h.

□

Remarque 9. *En calculant à l'aide d'un ordinateur, on peut obtenir les résultats suivants :*

- *sur les 256 réseaux de* $\mathrm{F}(2, 2)$*, 230 peuvent être simulés sans mémoire et 26 non (voir la figure 6.3) ;*
- *tous les réseaux de* $\mathrm{F}(3, 2)$ *peuvent être simulés sans mémoire ;*
- *sur les* $9^9 = 387\ 420\ 489$ *réseaux de* $\mathrm{F}(2, 3)$*, seulement* $2\ 102\ 400$ *ne peuvent pas être simulés sans mémoire.*

Il parait très difficile de calculer les cas où $n > 3$ *ou* $q > 3$ *car le nombre de réseaux à considérer* $((q^n)^{(q^n)})$ *explose rapidement.*

De plus, dans le cas général, la seule façon connue de générer l'ensemble des réseaux $h \in \mathrm{F}(n, q)$ *non simulables sans mémoire est la suivante. On calcule chaque ensemble* S_f *avec* $f \in \mathrm{F}(n, q)$ *et on soustrait leur union à* $\mathrm{F}(n, q)$*. Différentes symétries permettent de factoriser le temps de calcul de certains* S_f *mais ceci ne réduit pas significativement la difficulté du problème.*

FIGURE 6.3 – Chacun des 26 réseaux $h \in \mathrm{F}(2, 2)$ non simulables sans mémoire est une symétrie d'un des 4 réseaux représentés ici.

On va maintenant montrer que, pour tout alphabet non booléen et toute taille, il existe un réseau de cette taille et sur cet alphabet qui ne peut pas être simulé sans mémoire.

Théorème 28. *Pour tout* $n \geq 2$ *et* $q \geq 3$ *il existe* $h \in \mathrm{F}(n, q)$ *qui ne peut pas être simulé sans mémoire additionnelle.*

Démonstration. Notons $y = \mathtt{0}^{n-2}$, $a = \mathtt{00}y$, $b = \mathtt{10}y$, $c = \mathtt{01}y$, $d = \mathtt{21}y$. Considérons le réseau h qui calcule la chaine d'affectations $(d \to c \to b \to a)$. Autrement dit, $h(b) = h(a) = a$, $h(c) = b$, $h(d) = c$ et, pour tout $x \in \mathbb{A}^n \setminus \{a, b, c, d\}$, $h(x) = x$. On remarque que le rang de h est $q^n - 1$.

Supposons par l'absurde qu'il existe un réseau $f \in \mathrm{F}(n, q)$ et un mot $w \in [n]^*$ tel que $f^w = h$.

Premièrement, prouvons que $f^{(1)}$ est une instruction singulière et que la coordonnée 1 apparait dans w. On sait que h n'est pas bijectif. Par conséquent, il existe une étape de temps t telle que la fonction $f^{w_1 \cdots w_t}$ est singulière. Considérons la plus petite telle étape de temps t. Seule la paire de configurations $\{a, b\}$ a une unique image par h. Par conséquent, on a $f^{w_1 \cdots w_t}(a) = f^{w_1 \cdots w_t}(b)$. Posons $i = w_t$, $a' = f^{w_1,\ldots,w_{t-1}}(a)$ et $b' = f^{w_1,\ldots,w_{t-1}}(b)$. Par minimalité de t, on sait que $a' \neq b'$ et que i n'apparait pas dans $w_1 \ldots w_{t-1}$ (car l'instruction $f^{(i)}$ est singulière). Si $i \neq 1$, alors on a $a_i = b_i$ et on en déduit que $a'_i = b'_i$. Comme $a' \neq b'$, il existe $j \neq i = w_t$ tel que $a'_j \neq b'_j$. Mais alors $f_j^{w_1,\ldots,w_t}(a) = a'_j \neq b_j = f_j^{w_1,\ldots,w_t}(b)$ ce qui est faux. On a donc bien $i = w_t = 1$.

Notons $a' = f^{w_1 \cdots w_{t-1}}(a)$ et $b' = f^{w_1 \cdots w_{t-1}}(b)$. On remarque que $a'_1 = \mathtt{0}$ et $b'_1 = \mathtt{1}$ (car la coordonnée 1 n'est pas mise à jour dans $w_1 \ldots w_{t-1}$). Montrons que toute configuration $x \in \mathbb{A}^n$, avec $x_1 \neq d_1$, a une pré-image par $f^{(1)}$. Supposons par l'absurde que ce ne soit pas le cas pour une certaine configuration x. Prenons t' la dernière étape à laquelle la coordonnée 1 apparait dans w. Même si $f^{w_1 \cdots w_{t'-1}}$ était bijectif, le nombre de configurations $y \in f^{w_1 \cdots w_{t'}}(\mathbb{A}^n) \subseteq f^{(1)}(\mathbb{A}^n)$ telles que $y_1 = x_1$ serait d'au plus $q^{n-1} - 1$. Comme la coordonnée 1 n'est plus mise à jour après l'étape de temps t', le nombre d'images y de f^w telles que l'égalité $y_1 = x_1$ est respectée est inférieure à celui de la fonction h, ce qui est absurde. Par conséquent, toute configuration $x \in \mathbb{A}^n$, avec $x_1 \neq d_1$, doit avoir une pré-image par $f^{(1)}$.

Montrons maintenant que la coordonnée 1 n'apparait qu'une fois dans w. Supposons le contraire par l'absurde. Notons t' la deuxième étape de temps à laquelle la coordonnée 1 est mise à jour. Comme on vient de le voir, $f^{w_1 \cdots w_t}$ a un rang d'exactement $q^n - 1$ (tout comme h). Par conséquent, $f^{w_1 \cdots w_{t'-1}}$ doit avoir un rang de $q^n - 1$ (sinon on n'a pas l'égalité $f^w = h$). De plus, comme chaque configuration $x \in \mathbb{A}^n$ avec $x_1 \in \{0, 1\}$ a une pré-image par $f^{(i)}$, a' et b' ont une pré-image par $f^{w_1 \cdots w_{t'-1}}$. Notons ces pré-images a'' et b'' et remarquons que $\{a'', b''\} \neq \{a, b\}$ car a et b ont la même image par $f^{w_1 \cdots w_t}$ et donc par $f^{w_1 \cdots w_{t'-1}}$. On a

$$f^{w_1 \cdots w_{t'}}(a'') = f^{(1)}(a') = f^{(1)}(b') = f^{w_1 \cdots w_{t'}}(b'').$$

De ce fait, $h(a'') = h(b'')$ alors que $\{a'', b''\} \neq \{a, b\}$. Ceci est une contradiction. Par conséquent la coordonnée 1 n'apparait qu'une seule fois dans w.

Ceci signifie que, pour tout $x \in \mathbb{A}^n$, $f_1^{w_1 \cdots w_t}(x) = h_1(x)$. On en déduit que, $f^{(1)}(a') = f^{(1)}(b') = a'$. On a vu que b' doit avoir une pré-image par $f^{(1)}$. La seule configuration $x \in \mathbb{A}^n$ telle que $x_1 \neq b_1$ et $h_1(x) = b_1$ est c. Notons $c' = f_1^{w_1 \cdots w_{t-1}}(c)$. On a donc $f^{(1)}(c') = b'$

et donc $c'_{[2,n]} = b'_{[2,n]} = a'_{[2,n]}$. Or, $a'_1 = c'_1$ et donc $a' = c'$ ce qui pose une contradiction. Par conséquent, $f^w \neq h$ et le réseau $h \in \mathrm{F}(n,q)$ ne peut pas être simulé sans mémoire additionnelle.

□

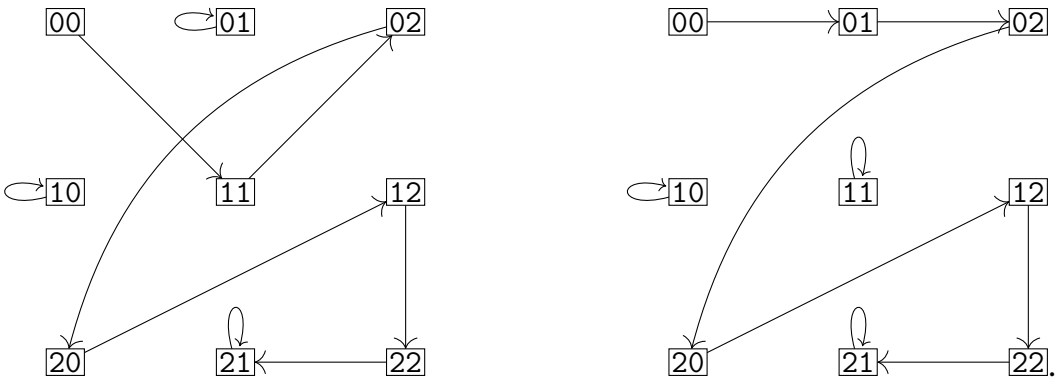
On conclut cette section par la conjecture suivante (dont la validité pour $n = 3$ a été obtenue à l'aide d'un ordinateur, comme indiqué dans la remarque 9).

Conjecture 5. *Pour tout $n \geq 3$ et, pour tout réseau $h \in \mathrm{F}(n,2)$, il existe un réseau $f \in \mathrm{F}(n,2)$ qui peut simuler h.*

6.7 Conclusions et perspectives

La plupart des questions ouvertes concernent la section 6.6. Dans celle-ci, on étudie des réseaux qui ne sont pas simulables sans mémoire additionnelle. On y montre qu'à part dans le cas trivial ou $n = 1$ ou quand $q = 2$, on peut toujours trouver un réseau non simulable dans $\mathrm{F}(n,q)$. La question la plus naturelle est surement de se demander s'il existe des conditions nécessaires et suffisantes facilement testables pour savoir si une fonction est simulable.

Il n'est pas clair qu'il y ait des conditions suffisantes faciles à vérifier systématiquement. Par exemple, si on s'intéresse aux deux réseaux de $\mathrm{F}(2,3)$ suivants :



On peut vérifier par ordinateur que le réseau représenté à gauche n'est pas simulable sans mémoire additionnelle alors que celui de droite, pourtant très ressemblant, l'est. Cet exemple permet de réfuter plusieurs hypothèses naïves sur une caractérisation des réseaux non simulables sans mémoire additionnelle. On ne peut donc pas se contenter de regarder le rang, les configurations avec la même image ou les configurations sans pré-image pour déterminer si un réseau est simulable.

Il est en revanche surement possible de trouver des conditions nécessaires pour ne pas être simulable. On pourrait peut-être alors prouver que tout réseau h dans $\mathrm{F}(n, 2)$ avec $n \geq 3$ échappe à ces conditions ce qui prouverait notre conjecture.

On pourrait aussi se poser des questions sur le temps minimum pour simuler un réseau sans mémoire dans le pire des cas. Cette question a un lien étroit avec la question de la possibilité d'écrire un certificat pour prouver qu'une fonction est séquentialisable.

Plusieurs restrictions du problème pourraient être intéressantes à étudier. Notamment, en bornant le degré des graphes d'interactions des graphes considérés ou en restreignant les fonctions locales à des fonctions monotones ou algébriques par exemple.

Chapitre 7

Conclusion et perspective

Dans l'introduction, nous nous sommes interrogés sur les rôles respectifs du graphe d'interaction, du mode de mise à jour et de l'alphabet d'un réseau d'automates sur sa dynamique. Nous allons maintenant voir comment le travail présenté tout au long de cette thèse nous renseigne sur ces questions et quel travail pourrait être réalisé pour consolider ces résultats.

Comme nous l'avons vu dans l'introduction, les cycles limites sont des éléments très étudiés de la dynamique des réseaux d'automates, entre autres parce qu'ils ont une interprétation biologique relativement immédiate. Dans ce cadre, un exercice classique est d'essayer de déduire des informations sur les cycles limites d'un réseau à l'aide de son graphe d'interaction (éventuellement signé) car c'est une information qui peut être obtenue, au moins partiellement, de manière expérimentale. Le chapitre 3 s'intègre dans ce cadre : nous nous y posons la question de la complexité algorithmique de faire de telles déductions. Ceci est une question très vaste, que nous avons abordée en supposant certaines hypothèses.

Premièrement, tous les problèmes abordés portaient sur les points fixes. Les points fixes sont des cycles limites particuliers : ils ne dépendent pas du mode de mise à jour bloc-séquentiel utilisé. Par contre, on pourrait s'intéresser à des problèmes sur les cycles limites en général. Dans ce cadre, le problème suivant pourrait être intéressant à étudier.

Problème de longueur des cycles limites
Entrée: Un graphe d'interaction signé G et un entier k
Sortie: Existe-t-il un réseau $f \in \mathrm{F}(G)$ et un mode de mise à jour bloc-séquentiel W tel que f^W a un cycle limite de taille k ?

Deuxièmement, nous avons supposé que le graphe d'interaction étudié était signé. Une extension naturelle de ces travaux serait donc d'étudier des graphes non signés. On pourrait aussi s'autoriser à ne pas prendre en compte tous les arcs du digraphe G donné en paramètre, en considérant tous les réseaux f dont le graphe d'interaction est un sous-graphe de G. Dans ce cadre, un grand nombre de questions deviennent beaucoup plus faciles. Par exemple, avec ces deux variantes du problème, on peut toujours associer un réseau avec au moins un point fixe à notre graphe. De plus, s'il est non signé, on peut lui associer un réseau sans aucun point fixe dès lors qu'il n'est pas acyclique. Il serait quand même intéressant de voir comment la difficulté de ces problèmes évolue en fonction du nombre de points fixes demandés. En particulier, dans le *network coding*, le digraphe G donné est non signé, on s'autorise à enlever des arcs et on se demande s'il existe une fonction avec au moins 2^τ points fixes.

Finalement, tout le long de ce chapitre nous avons supposé que l'alphabet du réseau considéré était booléen. À quel point peut-on généraliser tous ces résultats à des alphabets non booléens ? Soulignons d'abord que le graphe d'interaction signé d'un réseau booléen nous apprend beaucoup de choses sur ses fonctions locales, surtout quand son degré est limité. Par exemple, on sait que si en entrée d'un automate, il n'y a qu'un arc

positif, alors il n'a qu'une seule fonction locale possible. De même, s'il n'a que deux arcs entrants positifs alors il ne peut calculer que deux fonctions : OR et AND. À l'inverse, dans la même situation mais avec un alphabet de taille 3, il peut être associé à huit fonctions locales différentes s'il a un unique arc entrant positif et 172 s'il en a deux. De plus, de nombreuses propriétés élémentaires sur les réseaux booléens ne sont plus vraies sur les réseaux non booléens. Par exemple, si un graphe d'interaction a un cycle négatif isolé alors tout réseau booléen admis par ce graphe n'a aucun point fixe. Ce n'est pas vrai si on prend un alphabet de taille 3. En effet, considérons le graphe d'interaction signé G constitué d'une simple boucle négative, et le réseau $f : \{0,1,2\} \to \{0,1,2\}$ tel que $2 \xrightarrow{f} 0 \xrightarrow{f} 1 \xrightarrow{f} 1$. On voit que $f \in \mathrm{F}(G,3)$ (car $f(2) < f(1) \leq f(0)$) et pourtant f a bien un point fixe. En fait, on peut montrer que, pour tout graphe d'interaction signé fortement connexe et toute taille d'alphabet $q \geq 3$, on peut trouver $f \in \mathrm{F}(G,q)$ qui a au moins un point fixe et aucun autre cycle limite [66]. Maintenant, un certain nombre de propriétés restent vraies sur des alphabets non booléens. C'est notamment le cas pour le théorème 1 sur les graphes d'interaction acycliques, les théorèmes 2 et 3 sur les graphes d'interactions avec respectivement uniquement des cycles négatifs ou positifs et enfin le théorème 4 sur le *feedback vertex set*. Il serait intéressant d'étudier les propriétés qui sont conservées par le changement de taille de l'alphabet et voir si on peut adapter nos constructions pour obtenir des résultats de complexité sur les réseaux d'automates non booléens.

Dans le même ordre d'idée, dans le chapitre 4, nous avons caractérisé les graphes d'interaction qui admettent des réseaux d'automates expansifs. En effet, comme souligné dans le théorème 13, ce sont exactement ceux qui sont couvrables et connexes.

Dans ce chapitre, on a beaucoup mis en avant le fait que la taille de l'alphabet était très importante. En effet, dans la section 4.5, on a montré qu'on peut construire des graphes qui n'admettent des réseaux expansifs que sur des alphabets de tailles arbitrairement grandes. On a aussi présenté un graphe qui admet un réseau expansif pour tout alphabet mais pas d'expansif linéaire pour une infinité de tailles d'alphabet (voir la proposition 6). Pour finir, dans la section 4.7, on considère une notion plus forte d'expansivité et on prouve qu'il existe un tel réseau si et seulement si le graphe d'interaction est complet (voir le théorème 18), mais qu'il requiert un alphabet de taille quadratique par rapport à la taille du graphe (voir le corolaire 7).

En revanche, ce que l'on n'a pas fait, et qui pourrait être intéressant, est de faire varier le mode de mise à jour. En effet, on a considéré que le réseau était mis à jour parallèlement. Pourtant, étant donné un graphe G et $W = (W_1, \ldots, W_p)$ un mode de mise à jour bloc-séquentiel, on peut se poser la question suivante : est-ce qu'il existe $f \in \mathrm{F}(G)$ tel que f^W est expansif ? En utilisant le lemme 16 et en adaptant le théorème 11, on peut montrer que si f^W est expansif alors G doit être fortement connexe et pour tout $i \in [p]$, le graphe G restreint à W_i doit être couvrable. En particulier, si W est un mode de mise à jour séquentiel, alors G doit être boucle-complet. Il serait intéressant de vérifier si c'est une condition nécessaire et suffisante. De plus, pour rejoindre la problématique entre

séquentiel et parallèle abordée dans le chapitre 5 et en mettant de côté le graphe d'interaction, on pourrait se poser la question suivante. Est-ce que le caractère redondant de l'information des réseaux expansifs fait d'eux des réseaux plus faciles à séquentialiser ?

Les chapitres 5 et 6 étaient particulièrement intégrés dans la problématique de la variation du mode de mise à jour. On a montré dans le chapitre 5 qu'un mode de mise à jour séquentiel ne pouvait pas produire toutes les dynamiques possibles à l'inverse d'un mode de mise à jour parallèle. Plus précisément, on a montré qu'on pouvait créer des réseaux mis à jour en parallèle qui ne sont simulables que par des réseaux mis à jour en séquentiel avec environ une fois et demi plus d'automates. En outre, on a montré que c'était un résultat asymptotiquement optimal.

En cherchant des réseaux qui ont un coût de séquentialisation maximal on a aussi étudié l'importance de la taille de l'alphabet. Il semblerait que plus l'alphabet est grand, plus on peut trouver des réseaux avec un plus grand coût de séquentialisation. En outre, il semblerait qu'il y ait un saut concernant le coût de séquetialisation lorsque l'on passe d'un alphabet de taille 3 à un alphabet de taille 4 comme souligné dans le théorème 21.

On a montré dans le chapitre 6 que le mode de mise à jour séquentiel avec répétition était beaucoup plus complexe que le mode de mise à jour séquentiel normal. On peut trouver des réseaux $f \in \mathrm{F}(n+1, q)$ qui sont n-complets, c'est-à-dire des réseaux simulant tous les réseaux $h \in F(n, q)$ avec seulement un automate additionnel. On a montré des résultats similaires en augmentant la taille de l'alphabet plutôt que celle du réseau. En revanche, on a montré que, pour tous les alphabets non booléens, il existe des réseaux absolument non simulables séquentiellement sans mémoire additionnelle. La question booléenne reste ouverte, mais on pense que tout réseau d'automates booléen est simulable sans mémoire quand il est suffisamment grand.

Au-delà des modes de mise à jour séquentiels avec répétition, on pourrait considérer des modes de mise à jour bloc-séquentiel dégénérés $W = (W_1, \ldots, W_p)$ où les blocs W_i mis à jour ne sont pas forcément disjoints. En d'autres termes, une généralisation des séquentiels avec répétition. Pour tout réseau $h \in \mathrm{F}(n, q)$ avec $q \geq 3$, on pourrait alors se demander à partir de quel entier $1 \leq k \leq n$, il existe un réseau $f \in \mathrm{F}(n, q)$ et un mode de mise à jour bloc-séquentiel dégénéré W dont le plus gros bloc est de taille k et tels que $f^W = h$. On sait déjà que c'est possible pour $k = n$ mais est-ce que c'est vrai pour $k = n - 1$ quand n est suffisamment grand ? De manière encore plus forte, est-ce qu'en prenant k une constante suffisamment grande cela serait vrai pour toute taille n ?

Enfin, dans le chapitre 5, le graphe d'interaction a été abordé mais seulement pour donner des bornes sur le coût de séquentialisation d'un réseau. En revanche, aussi bien dans le chapitre 5 que dans le 6, les graphes d'interaction des réseaux qui simulent ont toujours été arbitraires et probablement quasi-complets. Une contrainte naturelle serait de s'assurer que le graphe d'interaction du réseau qui simule soit plus « creux ». Une question qui aurait probablement plus d'applications est la suivante : étant donné un réseau $h \in \mathrm{F}(n, q)$ et son graphe d'interaction G, est-ce qu'il existe un réseau $f \in \mathrm{F}(G, q')$

tel que f^w simule h (potentiellement en utilisant un alphabet plus gros) avec w un mode de mise à jour avec ou sans répétition ?

Pour terminer, pendant toute cette thèse nous avons considéré que chaque automate d'un même réseau prenait le même nombre d'états. Il serait intéressant de savoir quels sont les résultats présentés dans cette thèse qui restent valides quand un alphabet spécifique est associé à chaque automate du réseau.

Bibliographie

[1] Alfred Aho, John Hopcroft, and Jeffrey Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, 1974.

[2] Tatsuya Akutsu, Miyano Satoru, and Kuhara Satoru, *Identification of genetic networks from a small number of gene expression patterns under the Boolean network model*, Proceedings of PSB'99, 1999, pp. 17–28.

[3] Réka Albert, *Boolean modeling of genetic regulatory networks*, Complex networks, Springer, 2004, pp. 459–481.

[4] Noga Alon, *Asynchronous threshold networks*, Graphs and Combinatorics **1** (1985), no. 1, 305–310.

[5] Julio Aracena, *Maximum number of fixed points in regulatory Boolean networks*, Bulletin of Mathematical Biology **70** (2008), no. 5, 1398–1409.

[6] Julio Aracena, Jacques Demongeot, Eric Fanchon, and Marco Montalva, *On the number of different dynamics in Boolean networks with deterministic update schedules*, Mathematical Biosciences **242** (2013), 188–194.

[7] _____, *On the number of update digraphs and its relation with the feedback arc sets and tournaments*, Discrete Applied Mathematics **161** (2013), no. 10-11, 1345–1355.

[8] Julio Aracena, Jacques Demongeot, and Eric Goles, *Fixed points and maximal independent sets in AND–OR networks*, Discrete Applied Mathematics **138** (2004), no. 1, 277–288.

[9] _____, *Positive and negative circuits in discrete neural networks*, IEEE Transactions on Neural Networks **15** (2004), no. 1, 77–83.

[10] Julio Aracena, Eric Fanchon, Marco Montalva, and Mathilde Noual, *Combinatorics on update digraphs in boolean networks*, Discrete Applied Mathematics **159** (2011), no. 6, 401–409.

[11] Julio Aracena, Eric Goles, Andrés Moreira, and Lilian Salinas, *On the robustness of update schedules in Boolean networks*, Biosystems **97** (2009), no. 1, 1–8.

[12] Julio Aracena, Luis Gómez, and Lilian Salinas, *Limit cycles and update digraphs in boolean networks*, Discrete Applied Mathematics **161** (2013), no. 1-2, 1–12.

[13] Julio Aracena, Adrien Richard, and Lilian Salinas, *Maximum number of fixed points in AND-OR-NOT networks*, Journal of Computer and System Sciences **80** (2014), no. 7, 1175–1190.

[14] _____, *Fixed points in conjunctive networks and maximal independent sets in graph contractions*, Journal of Computer and System Sciences **88** (2017), 143–163.

[15] _____, *Number of fixed points and disjoint cycles in monotone Boolean networks*, SIAM Journal on Discrete Mathematics **31** (2017), no. 3, 1702–1725.

[16] Pablo Arrighi and Gilles Dowek, *Causal graph dynamics*, Proceedings of ICALP'12, 2012, pp. 54–66.

[17] Jorgen Bang-Jensen and Gregory Gutin, *Digraphs: Theory, algorithms and applications*, Springer, 2009.

[18] Elwyn Berlekamp, *Algebraic coding theory - revised edition*, World Scientific, 2015.

[19] François Blanchard and Alejandro Maass, *Dynamical properties of expansive one-sided cellular automata*, Israel Journal of Mathematics **99** (1997), no. 1, 149–174.

[20] John Adrian Bondy and Uppaluri Murty, *Graph theory*, 1st ed., Springer, 2008.

[21] Mike Boyle, *Open problems in symbolic dynamics*, Contemporary Mathematics **469** (2008), 69–118.

[22] Mike Boyle and Bryant Lee, *Jointly periodic points in cellular automata: computer explorations and conjectures*, Experimental Mathematics **16** (2007), no. 3, 293–302.

[23] Mike Boyle and Douglas Lind, *Expansive subdynamics*, Transactions of the American Mathematical Society **349** (1997), no. 1, 55–102.

[24] Florian Bridoux, Alonso Castillo-Ramirez, and Maximilien Gadouleau, *Complete simulation of automata networks*, arXiv (2018), 1504.00169.

[25] Florian Bridoux, Nicolas Durbec, Kévin Perrot, and Adrien Richard, *Complexity and fixed points in Boolean networks*, Proceedings of CiE'19, 2019, Accepted.

[26] Florian Bridoux, Maximilien Gadouleau, and Guillaume Theyssier, *Expansive automata networks*, arXiv (2019), 1902.08007, In review.

[27] Florian Bridoux, Pierre Guillon, Kévin Perrot, Sylvain Sené, and Guillaume Theyssier, *On the cost of simulating a parallel boolean automata network by a block-sequential one*, Proceedings of TAMC'17, 2017, pp. 112–128.

[28] Serge Burckel, *Closed iterative calculus*, Theoretical Computer Science (1996), no. 1-2, 371–378.

[29] Serge Burckel, *Elementary decompositions of arbitrary maps over finite sets*, Journal of Symbolic Computation **37** (2004), no. 3, 305–310.

[30] Serge Burckel, Emeric Gioan, and Emmanuel Thomé, *Mapping computation with no memory*, Proceedings of UC'09, 2009, pp. 85–97.

[31] _____, *Computation with no memory, and rearrangeable multicast networks*, Discrete Mathematics and Theoretical Computer Science **Vol. 16** (2014), no. 1, 121–142.

[32] Serge Burckel, Emeric Gioan, and Emmanuel Thomé, *Computation with no memory, and rearrangeable multicast networks*, Discrete Mathematics and Theoretical Computer Science **16** (2014), 121–142.

[33] Serge Burckel and Marianne Morillon, *Three generators for minimal writing-space computations*, RAIRO-Theoretical Informatics and Applications **34** (2000), no. 2, 131–138.

[34] ———, *Quadratic sequential computations of Boolean mappings*, Theory of Computing Systems **37** (2004), no. 4, 519–525.

[35] ———, *Sequential computation of linear Boolean mappings*, Theoretical Computer Science **314** (2004), no. 1-2, 287–292.

[36] Kenneth Bush et al., *Orthogonal arrays of index unity*, The Annals of Mathematical Statistics **23** (1952), no. 3, 426–434.

[37] Steve Butler, Mohammad Hajiaghayi, Robert Kleinberg, and Tom Leighton, *Hat guessing games*, SIAM review **51** (2009), no. 2, 399–413.

[38] Peter Cameron, Ben Fairbairn, and Maximilien Gadouleau, *Computing in matrix groups without memory*, Chicago Journal of Theoretical Computer Science (2014), no. 08, 1–16.

[39] ———, *Computing in permutation groups without memory*, Chicago Journal of Theoretical Computer Science **2014** (2014), no. 7, 1–20.

[40] Tullio Ceccherini-Silberstein and Michel Coornaert, *Cellular automata and groups*, Springer, 2010.

[41] Soshin Chikazumi and Chad Graham, *Physics of ferromagnetism*, 2nd ed., Oxford University Press on Demand, 2009.

[42] Philip Chou, Yunnan Wu, and Kamal Jain, *Practical network coding*, Proceedings of ALLERTON'03, 2003, invited paper, pp. 40–49.

[43] Samuel Collombet, Chris van Oevelen, Jose Ortega, Wassim Abou-Jaoudé, Bruno Di Stefano, Morgane Thomas-Chollier, Thomas Graf, and Denis Thieffry, *Logical modeling of lymphoid and myeloid cell specification and transdifferentiation*, Proceedings of the National Academy of Sciences **114** (2017), no. 23, 5792–5799.

[44] Paul Cull, *Linear analysis of switching nets*, Kybernetik **8** (1971), no. 1, 31–39.

[45] Hidde De Jong, *Modeling and simulation of genetic regulatory systems: a literature review*, Journal of computational biology **9** (2002), no. 1, 67–103.

[46] Franck Delaplace, Hanna Klaudel, Tarek Melliti, and Sylvain Sené, *Analysis of modular organisation of interaction networks based on asymptotic dynamics*, Proceedings of CMSB'12, 2012, pp. 148–165.

[47] Jacques Demongeot, Adrien Elena, Mathilde Noual, Sylvain Sené, and Florence Thuderoz, *"Immunetworks", intersecting circuits and dynamics*, Journal of Theoretical Biology **280** (2011), no. 1, 19–33.

[48] Jacques Demongeot, Eric Goles, Michel Morvan, Mathilde Noual, and Sylvain Sené, *Attraction basins as gauges of robustness against boundary conditions in biological complex systems*, PloS one **5** (2010), no. 8, e11793.

[49] Roland L'vovich Dobrushin, *The existence of a phase transition in the two-and three-dimensional ising models*, Teoriya Veroyatnostei i ee Primeneniya **10** (1965), no. 2, 209–230.

[50] _____, *Gibbsian random fields for lattice systems with pairwise interactions*, Functional Analysis and its Applications **2** (1968), no. 4, 292–301.

[51] _____, *The problem of uniqueness of a Gibbsian random field and the problem of phase transitions*, Functional Analysis and its Applications **2** (1968), no. 4, 302–312.

[52] Pál Dömösi and Chrystopher Nehaniv, *Algebraic theory of automata networks: An introduction*, SIAM, 2005.

[53] Bruno Durand and Zsuzsanna Róka, *The game of life: universality revisited*, Cellular automata, Springer, 1999, pp. 51–74.

[54] Jack Edmonds, *Systems of distinct representatives and linear algebra*, Journal of Research of the National Bureau of Standards **71B** (1967), 241–245.

[55] John Ellis, Ivan Sudborough, and Jonathan Turner, *The vertex separation and search number of a graph*, Information and Computation **113** (1994), no. 1, 50–79.

[56] Drew Endy and Roger Brent, *Modelling cellular behaviour*, Nature **409** (2001), no. 6818, 391.

[57] Zoltán Ésik, *A note on isomorphic simulation of automata by networks of two-state automata*, Discrete Applied Mathematics **30** (1991), no. 1, 77–82.

[58] Tomás Feder, *A new fixed point approach for stable networks and stable marriages*, Journal of Computer and System Sciences **45** (1992), no. 2, 233–284.

[59] Patrik Floréen and Pekka Orponen, *Attraction radii in binary hopfield nets are hard to compute*, Neural Computation **5** (1993), no. 5, 812–821.

[60] Francoise Fogelman-Soulie, *Parallel and sequential computation on Boolean networks*, Theoretical computer science **40** (1985), 275–300.

[61] Pierre Fraigniaud, Marc Heinrich, and Adrian Kosowski, *Local conflict coloring*, Proceedings of FOCS'16, 2016, pp. 625–634.

[62] Pierre Fraigniaud, Amos Korman, and David Peleg, *Local distributed decision*, Proceedings of FOCS'11, 2011, pp. 708–717.

[63] Maximilien Gadouleau, *Finite dynamical systems, hat games, and coding theory*, SIAM Journal on Discrete Mathematics **32** (2018), no. 3, 1922–1945.

[64] _____, *On the influence of the interaction graph on a finite dynamical system*, Natural Computing (2018), 1–14, (accepted) https://doi.org/10.1007/s11047-019-09732-y.

[65] Maximilien Gadouleau, *On the rank and periodic rank of finite dynamical systems*, The Electronic Journal of Combinatorics **25** (2018), no. 3, 1–16.

[66] Maximilien Gadouleau and Adrien Richard, *Simple dynamics on graphs*, Theoretical Computer Science **628** (2016), 62–77.

[67] Maximilien Gadouleau, Adrien Richard, and Søren Riis, *Fixed points of Boolean networks, guessing graphs, and coding theory*, SIAM Journal on Discrete Mathematics **29** (2015), no. 4, 2312–2335.

[68] Maximilien Gadouleau and Søren Riis, *Memoryless computation: new results, constructions, and extensions*, Theoretical Computer Science **562** (2015), 129–145.

[69] Maximilien Gadouleau and Søren Riis, *Graph-theoretical constructions for graph entropy and network coding based communications*, IEEE Transactions on Information Theory **57** (2011), no. 10, 6703–6717.

[70] Martin Gardner, *The fantastic combinations of John Conway's new solitaire game "life"*, Scientific American **223** (1970), 120–123.

[71] Carlos Gershenson, *Classification of random Boolean networks*, Proceedings of ICAL'03, 2003, pp. 1–8.

[72] Eric Goles and Servet Martínez, *Neural and automata networks: dynamical behavior and applications*, Kluwer Academic Publishers, 1990.

[73] Eric Goles and Mathilde Noual, *Block-sequential update schedules and Boolean automata circuits*, Proceedings of AUTOMATA'10, 2010, pp. 41–50.

[74] Eric Goles and Lilian Salinas, *Comparison between parallel and serial dynamics of Boolean networks*, Theoretical Computer Science **396** (2008), no. 1-3, 247–253.

[75] ———, *Sequential operator for filtering cycles in Boolean networks*, Advances in Applied Mathematics **45** (2010), no. 3, 346–358.

[76] Eric Goles and Maurice Tchuente, *Iterative behaviour of generalized majority functions*, Mathematical Social Sciences **4** (1983), no. 3, 197–204.

[77] David Green, Tania Leishman, and Suzanne Sadedin, *The emergence of social consensus in Boolean networks*, Proceedings of ALife'2018, 2007, pp. 402–408.

[78] Inman Harvey and Terry Bossomaier, *Time out of joint: Attractors in asynchronous random Boolean networks*, Proceedings of ECAL'97, 1997, pp. 67–75.

[79] Samad Hedayat, Neil Sloane, and John Stufken, *Orthogonal arrays*, Springer-Verlag, 1999.

[80] Tracey Ho, Muriel Médard, Ralf Koetter, David Karger, Michelle Effros, Jun Shi, and Ben Leong, *A random linear network coding approach to multicast*, IEEE Transactions on Information Theory **52** (2006), no. 10, 4413–4430.

[81] Sui Huang and Stuart Kauffman, *Complex gene regulatory networks–from structure to biological observables: cell fate determination*, Computational complexity: theory, techniques, and applications (2012), 527–560.

[82] Ernst Ising, *Beitrag zur theorie des ferromagnetismus*, Zeitschrift für Physik A Hadrons and Nuclei **31** (1925), no. 1, 253–258.

[83] Joonatan Jalonen and Jarkko Kari, *On dynamical complexity of surjective ultimately right-expansive cellular automata*, Proceedings of AUTOMATA'18, 2018, pp. 57–71.

[84] Winfried Just and German Enciso, *Extremely chaotic Boolean networks*, arXiv e-prints (2008), arXiv:0811.0115.

[85] Winfried Just and Maciej Malicki, *Cooperative Boolean systems with generically long attractors I*, Journal of Difference Equations and Applications **19** (2013), no. 5, 772–795.

[86] _____, *Cooperative Boolean systems with generically long attractors II*, Advances in Difference Equations **19** (2013), no. 1, 268–291.

[87] Jarkko Kari, *Theory of cellular automata: A survey*, Theoretical computer science **334** (2005), no. 1-3, 3–33.

[88] Hisao Kato, *Expansive homeomorphisms on surfaces with holes*, Topology and its Applications **82** (1998), no. 1-3, 267–277.

[89] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft, *Xors in the air: practical wireless network coding*, IEEE/ACM Transactions on Networking (ToN) **16** (2008), no. 3, 497–510.

[90] Stuart Kauffman, *Metabolic stability and epigenesis in randomly constructed genetic nets*, Journal of Theoretical Biology **22** (1969), no. 3, 437–467.

[91] _____, *Cellular homeostasis, epigenesis and replication in randomly aggregated macromolecular systems*, Journal of Cybernetics **1** (1971), no. 1, 71–96.

[92] _____, *Origins of order self-organization and selection in evolution*, Oxford University Press, 1993.

[93] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein, *Random Boolean network models and the yeast transcriptional network*, Proceedings of the National Academy of Sciences **100** (2003), no. 25, 14796–14799.

[94] Marcelle Kaufman and René Thomas, *Model analysis of the bases of multistationarity in the humoral immune response*, Journal of Theoretical Biology **129** (1987), no. 2, 141–62.

[95] Marcelle Kaufman, Jacques Urbain, and René Thomas, *Towards a logical analysis of the immune response*, Journal of Theoretical Biology **114** (1985), no. 4, 527–561.

[96] Marcelline Kaufman, Christophe Soulé, and René Thomas, *A new necessary condition on interaction graphs for multistationarity*, Journal of Theoretical Biology **248** (2007), no. 4, 675–685.

[97] Sven Kosub, *Dichotomy results for fixed-point existence problems for Boolean dynamical systems*, Mathematics in Computer Science **1** (2008), no. 3, 487–505.

[98] Petr Kůrka, *Topological and symbolic dynamics*, Société Mathématique de France, 2003.

[99] Dietlinde Lau, *Function algebras on finite sets: Basic course on many-valued logic and clone theory*, Springer, 2006.

[100] Nicolas Le Novère, *Quantitative and logic modelling of molecular and gene networks*, Nature Reviews Genetics **16** (2015), no. 3, 146.

[101] Shuo-Yen Li, Raymond Yeung, and Ning Cai, *Linear network coding*, IEEE Transactions on Information Theory **49** (2003), no. 2, 371–381.

[102] Michael Littman, Judy Goldsmith, and Martin Mundhenk, *The computational complexity of probabilistic planning*, Journal of Artificial Intelligence Research **9** (1998), 1–36.

[103] Florence MacWilliams and Neil Sloane, *The theory of error-correcting codes*, North-Holland, 1978.

[104] Abibatou Mbodj, Hilary Gustafson, Lucia Ciglar, Guillaume Junion, Aitor Gonzalez, Charles Girardot, Laurent Perrin, Eileen EM Furlong, and Denis Thieffry, *Qualitative dynamical modelling can formally explain mesoderm specification and predict novel developmental phenotypes*, PLoS Computational Biology **12** (2016), no. 9, e1005073.

[105] William McCuaig, *Pólya's permanent problem*, Electronic Journal of Combinatorics **11** (2004), no. 1, 79.

[106] Warren McCulloch and Walter Pitts, *A logical calculus of the ideas immanent in nervous activity*, The Bulletin of Mathematical Biophysics **5** (1943), no. 4, 115–133.

[107] Tarek Melliti, Damien Regnault, Adrien Richard, and Sylvain Sené, *Asynchronous simulation of Boolean networks by monotone Boolean networks*, Proceedings of ACRI'18, 2016, pp. 182–191.

[108] Gary Mullen and Daniel Panario, *Handbook of finite fields*, Chapman and Hall/CRC, 2013.

[109] Masakazu Nasu, *Nondegenerate q-biresolving textile systems and expansive automorphisms of onesided full shifts*, Transactions of the American Mathematical Society **358** (2006), no. 2, 871–891.

[110] John Von Neumann, *Theory of self-reproducing automata*, University of Illinois Press, 1966.

[111] Mathilde Noual, *Updating automata networks*, Ph.D. thesis, Ecole normale supérieure de lyon, 2012.

[112] Mathilde Noual, Damien Regnault, and Sylvain Sené, *About non-monotony in Boolean automata networks*, Theoretical Computer Science **504** (2013), 12–25.

[113] Mathilde Noual and Sylvain Sené, *Synchronism versus asynchronism in monotonic Boolean automata networks*, Natural Computing **17** (2018), no. 2, 393–402.

[114] Matti Nykter, Nathan Price, Maximino Aldana, Stephen Ramsey, Stuart Kauffman, Leroy Hood, Olli Yli-Harja, and Ilya Shmulevich, *Gene expression dynamics in the macrophage exhibit criticality*, Proceedings of the National Academy of Sciences **105** (2008), no. 6, 1897–1900.

[115] Lars Onsager, *Crystal statistics. i. a two-dimensional model with an order-disorder transition*, Physical Review **65** (1944), no. 3-4, 117–149.

[116] Christos Papadimitriou, *Computational complexity*, Pearson, 1993.

[117] Christophe Papazian, *Graphes d'automates finis: Reconnaissance et calcul*, Ph.D. thesis, Ecole normale supérieure de lyon, 2002.

[118] Loïc Paulevé and Aadrien Richard, *Topological fixed points in boolean networks*, Comptes Rendus de l'Académie des Sciences-Series I-Math. **348** (2010), no. 15-16, 825–828.

[119] Sophie Piccard, *Sur les fonctions définies dans les ensembles finis quelconques*, Fundamenta mathematicae **1** (1935), no. 24, 298–301.

[120] Barry Pierce and Carol Wallace, *Differentiation of malignant to benign cells*, Cancer research **31** (1971), no. 2, 127–134.

[121] Marcus Pivato, *Positive expansiveness versus network dimension in symbolic dynamical systems*, Theoretical Computer Science (2011), no. 30, 3838–3855.

[122] Svatopluk Poljak and Miroslav Sůra, *On periodical behaviour in societies with symmetric influences*, Combinatorica **3** (1983), no. 1, 119–121.

[123] Élisabeth Remy, Paul Ruet, and Denis Thieffry, *Graphic requirements for multistability and attractive cycles in a Boolean dynamical framework*, Advances in Applied Mathematics **41** (2008), no. 3, 335–350.

[124] Adrien Richard, *Positive circuits and maximal number of fixed points in discrete dynamical systems*, Discrete Applied Mathematics **157** (2009), no. 15, 3281–3288.

[125] ———, *Fixed points and connections between positive and negative cycles in Boolean networks*, Discrete Applied Mathematics **243** (2018), 1–10.

[126] ———, *Positive and negative cycles in Boolean networks*, Journal of Theoretical Biology **463** (2019), 67–76.

[127] Søren Riis, *Utilising public information in network coding*, Tech. report, Citeseer, 2005.

[128] ———, *Information flows, graphs and their guessing numbers*, Proceedings of WiOpt'06, 2006, pp. 1–9.

[129] Søren Riis, *Graph entropy, network coding and guessing games*, arXiv e-prints (2007), arXiv:0711.4175.

[130] François Robert, *Blocs-h-matrices et convergence des méthodes iteratives classiques par blocs*, Linear Algebra and its Applications **2** (1969), no. 2, 223–265.

[131] _____, *Contraction en norme vectorielle: convergence d'itérations chaotiques pour des équations non linéaires de point fixe à plusieurs variables*, Linear Algebra and its Applications **13** (1976), no. 1-2, 19–35.

[132] _____, *Itérations sur des ensembles finis et automates cellulaires contractants*, Linear Algebra and its Applications **29** (1980), 393–412.

[133] _____, *Discrete iterations: a metric study*, Springer, 1986.

[134] Yves Robert and Maurice Tchuente, *Connection-graph and iteration-graph of monotone Boolean functions*, Discrete Applied mathematics **11** (1985), no. 3, 245–253.

[135] Neil Robertson, Paul Seymour, and Robin Thomas, *Permanents, pfaffian orientations, and even directed circuits*, Annals of Mathematics **150** (1999), no. 3, 929–975.

[136] David Ruelle, *Statistical mechanics of a one-dimensional lattice gas*, Communications in Mathematical Physics **9** (1968), no. 4, 267–278.

[137] _____, *Statistical mechanics: Rigorous results*, World Scientific, 1999.

[138] Gonzalo Ruz, Tania Timmermann, and Eric Goles, *Building synthetic networks of the budding yeast cell-cycle using swarm intelligence*, Proceeding of ICMLA'12, 2012, pp. 120–125.

[139] James Sakoda, *The checkerboard model of social interaction*, The Journal of Mathematical Sociology **1** (1971), no. 1, 119–132.

[140] Yara-Elena Sanchez-Corrales, Elena Alvarez-Buylla, and Luis Mendoza, *The arabidopsis thaliana flower organ specification gene regulatory network determines a robust differentiation process*, Journal of Theoretical Biology **264** (2010), no. 3, 971–983.

[141] Thomas Schelling, *Dynamic models of segregation*, Journal of Mathematical Sociology **1** (1971), no. 2, 143–186.

[142] Jean-Pierre Serre, *Cours d'arithmétique*, Puf, 1970.

[143] Christophe Soulé, *Mathematical approaches to differentiation and gene regulation*, Comptes Rendus Biologies **329** (2006), no. 1, 13–20.

[144] Françoise Fogelman Soulié, Yves Robert, and Maurice Tchuente, *Automata networks in computer science: Theory and applications*, Manchester University Press, 1987.

[145] Robert Tarjan, *Depth-first search and linear graph algorithms*, SIAM Journal on Computing **1** (1972), no. 2, 146–160.

[146] Alfred Tarski, *A lattice-theoretical fixpoint theorem and its applications.*, Pacific Journal of Mathematics **5** (1955), no. 2, 285–309.

[147] Maurice Tchuente, *Parallel calculation of a linear mapping on a computer network*, Linear Algebra and its Applications **28** (1979), 223–247.

[148] _____, *Parallel realization of permutations over trees*, Discrete Mathematics **39** (1982), no. 2, 211–214.

[149] _____, *Computation of Boolean functions on networks of binary automata*, Journal of Computer and System sciences **26** (1983), no. 2, 269–277.

[150] _____, *Permutation factorization on star-connected networks of binary automata*, SIAM Journal on Algebraic Discrete Methods **6** (1985), no. 3, 537–540.

[151] _____, *Computation on finite networks of automata*, Proceedings of LITP'86, 1986, pp. 53–67.

[152] _____, *Sequential simulation of parallel iterations and applications*, Theoretical Computer Science **48** (1986), 135–144.

[153] René Thomas, *Boolean formalization of genetic control circuits*, Journal of Theoretical Biology **42** (1973), no. 3, 563–585.

[154] _____, *On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations*, Numerical methods in the study of critical phenomena, Springer, 1981, pp. 180–193.

[155] René Thomas and Richard d'Ari, *Biological feedback*, CRC press, 1990.

[156] René Thomas and Marcelle Kaufman, *Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits*, Chaos **11** (2001), no. 1, 180–195.

[157] René Thomas, *Regulatory networks seen as asynchronous automata: A logical description*, Journal of Theoretical Biology **153** (1991), no. 1, 1–23.

[158] Laurent Tournier and Madalena Chaves, *Interconnection of asynchronous Boolean networks, asymptotic and transient dynamics*, Automatica **49** (2013), no. 4, 884–893.

[159] Stanislaw Ulam, *On some mathematical problems connected with patterns of growth of figures*, Proceedings of Symposia in Applied Mathematics vol XIV, 1962, pp. 215–226.

[160] Angela Wu and Azriel Rosenfeld, *Cellular graph automata. I. Basic concepts, graph property measurement, closure properties*, Information and Control **42** (1979), no. 3, 305–329.

[161] Angela Wu and Azriel Rosenfeld, *Cellular graph automata. II. Graph and subgraph isomorphism, graph structure recognition*, Information and Control **42** (1979), 330–353.

[162] Boting Yang and Yi Cao, *Digraph searching, directed vertex separation and directed pathwidth*, Discrete Applied Mathematics **156** (2008), no. 10, 1822–1837.